UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

The importance of input features on deep neural networks when predicting foreign exchange rates

Heinrich von Stein

Thesis presented in fulfilment of the requirements for the degree of
MCom (Operations Research)
in the Faculty of Economic and Management Sciences at Stellenbosch University

Supervisor: Prof SE Visagie

March 2021

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2021

Abstract

Data has become a form of currency that can govern both the success and failure of almost every business, individual or idea. Raw, unprocessed financial data form the basis for new discoveries in this thesis. Since financial data is non-stationary and filled with noise, careful consideration is required when selecting forecasting models.

Machine learning (ML) has grown from just a concept to a leading analysis/prediction tool used in almost every industry in the world. The immense volume of data generated, mined and collected is the fuel that keeps the interest and development of ML alive. Without data, ML would not be able to advance in the way that it has. However, a wealth of data does not imply that all data is relevant and/or important. Selecting input variables for ML models is vitally important.

The effect and importance of input features was investigated on three different neural network (NN) architectures: a LSTM model, and two hybrid CNN-LSTM and Multi-Head CNN-LSTM models. Using the prediction accuracy, MSE, adjusting the accuracy threshold and classification accuracy, a comparison was done between the different tests, which used different input features, and the overall best performing NNs. The input features that were tested included: the open, high, low, close, 9-day and 21-day moving averages, the price difference, Relative Strength Index, Heikin Ashi, Ichimoku Kinko CLoud, bollinger Bands, 3,6 and 9 month implied volatility and risk reversal, 1^{st} and 2^{nd} differences and features determined through principal component analysis. As NNs are sensitive to network architectures, several architectures were also investigated for each input feature, thus allowing the opportunity for each test to find a possible optimal configuration.

It was found that the Multi-Head model obtained the best overall prediction accuracy of 25.6% when Bollinger Bands were added to the baseline input features: open, high, low and closing price of the USD/ZAR exchange rate. However, the Multi-Head model was outperformed by Multiple Regression which obtained a prediction accuracy of 30.4% using features obtained through Principal Component Analysis — with a binary increase input feature having the greatest influence on predictions made.

Opsomming

Data het 'n vorm van geldeenheid geword wat die sukses en mislukking van byna elke onderneming, individu of idee kan bepaal. Rou, onverwerkte finalsiële data vorm die basis vir die nuwe ontdekkings wat in hierdie tesis gemaak word. Aangesien finansiële gegewens nie-stasionêr en gevul met geraas is, moet deeglike oorweging geskenk word aan die keuse van vooruitskattingsmodelle.

Masjienleer (ML) het gegroei van net 'n konsep tot 'n toonaangewende hulpmiddel vir analise/voorspelling wat in byna elke bedryf in die wêreld gebruik word. Die groot hoeveelheid data wat gegenereer, ontgin en versamel word, is die brandstof wat die belangstelling en ontwikkeling van ML lewendig hou. Sonder data sou ML nie kon vorder soos dit het nie. 'n Oorvloed data impliseer egter nie dat alle data relevant en/of belangrik is nie. Die keuse van invoerveranderlikes vir ML-modelle is van uiterste belang.

Die effek en belangrikheid van invoereienskappe is ondersoek in drie verskillende neurale netwerkargitekture (NN): 'n LSTM-model en twee hibriede CNN-LSTM- en meerkoppige CNN-LSTM-modelle. Met behulp van die maatstaf, gemiddelde fout kwadraat (MSE), die aanpassing van die akkuraatheidsdrempel en klassifikasie-akkuraatheid, is 'n vergelyking gedoen tussen die verskillende toetse, wat verskillende invoereienskappe gebruik, en die algehele presatsie van NN's. Die invoereienskappe wat getoets is, sluit in: die openings-, hoë, lae en sluitingsprys, 9-dae en 21-dae bewegende gemiddeldes, die prysverskil, Relatiewe Sterkte Indeks, Heikin Ashi, Ichimoku Kinko CLoud, Bollinger Bande, 3, 6 en 9 maand geïmpliseerde volatiliteit en risiko-omkering, 1^{ste} en 2^{de} verskille en kenmerke wat bepaal word deur hoofkomponentanalise. Aangesien NN's sensitief is vir netwerkargitekture, is daar ook ondersoek ingestel na verskeie argitekture vir elke stel invoereienskappe, wat vir elke toets die geleentheid bied om 'n moontlike optimale konfigurasie te vind.

Die meerkoppige model lewer die beste algehele voorspellingsakkuraatheid van 25.6%, toe Bollinger Bande by die basisinvoereienskappe gevoeg is, naamlik: openings-, hoë, lae en sluitingsprys van die USD/ZAR-wisselkoers. Die meerkoppige model is egter oortref deur meervoudige regressie, wat 'n voorspellingsakkuraatheid van 30.4% behaal het deur gebruik te maak van funksies wat verkry is deur hoofkomponentanalise — met 'n binêre toename-invoerfunksie wat die grootste invloed het op voorspellings wat gemaak is.

List of Acronyms

ANN	Artificial Neural Network
BPNN	Multi-layer Neural Network with Back-Propagation
CEFLANN	Computationally Efficient Functional Link Artificial Neural Network
CNN	Convolutional Neural Network
DRPNN	Dynamic Ridge Polynomial Neural Network
DT	Decision Tree
GA	Genetic Algorithm
GAN	Generative Adversarial Network
HA	Heiken-Ashi
KMA	Kaiser's Measure of Sampling Adequacy
KNN	K Nearest Neighbour
LSTM	Long Short-Term Memory
MA	Moving Average
MACD	Moving Average Convergence and Divergence
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MAR	Moving Average Reversion
ML	Machine Learning
MLP	Multilayer Perceptron
MR	Multiple regression
MRV	Mean Reversion
MSE	Mean Square Error
NMSE	Normalised Mean Square Error

NN	Neural Network
NSE	National Stock Exchange
NYSE	New York Stock Exchange
OTM	Out of money
PC	Principal component
PCA	Principal Component Analysis
PSNN	Pi-Sigma Neural Network
RBF	Radial Basis Function
RDP	Relative Difference in Percentage of price
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RPNN	Ridge Polynomial Neural Network
RSI	Relative Strength Index
ReLu	Rectified Linear Unit
S&P 500	Standard & Poor's 500
SAE	Stacked Autoencoder
SES	Simple Exponential Smoothing
SMA	Simple Moving Average
SVM	Support Vector Machine
SVR	Support Vector Regression
USA	United States of America
USD	United States Dollar
WR	Larry William's R%
WT	Wavelet Transform
ZAR	South African Rand

Contents

List of Acronyms **vii**

1 Introduction **1**

1.1 Foreign exchange . . . 2
1.2 Machine learning . . . 3
1.3 Problem description . . . 6
1.4 Thesis scope . . . 6
1.5 Thesis objectives . . . 6
1.6 Thesis layout . . . 6

2 Literature review **9**

2.1 Foreign exchange . . . 9
2.2 Stock market . . . 11
2.3 Conclusion . . . 13

3 Data and modelling **15**

3.1 Features . . . 15
3.1.1 Moving average . . . 16
3.1.2 Bollinger bands . . . 16
3.1.3 Relative strength index . . . 18
3.1.4 Heiken-Ashi candlesticks . . . 19
3.1.5 Ichimoku kinki cloud . . . 20
3.1.6 Implied volatility and risk reversal . . . 20
3.2 Convolutional neural network . . . 22
3.3 Long-short term memory . . . 23
3.4 Data processing . . . 25
3.5 Modelling . . . 26
3.5.1 LSTM network . . . 27

3.5.2 CNN-LSTM network 28
3.5.3 Multi-head CNN-LSTM network 28

4 Principal component analysis 31
4.1 PCA criterion 31
4.2 Applying PCA 32

5 Statistical forecasting models 35
5.1 Naïve 35
5.2 Simple exponential smoothing 36
5.3 Multiple regression 36
5.4 Naïve results 37
5.5 Simple exponential smoothing results 37
5.6 Multiple regression results 39

6 Results for the LSTM model 47
6.1 Results for experiments in Group A 47
6.2 Results for experiments in Group B 52

7 Results for the CNN-LSTM model 57
7.1 Results for experiments in Group A 57
7.2 Results for experiments in Group B 61

8 Results for the Multi-head CNN-LSTM network 67
8.1 Results for experiments in Group A 67
8.2 Results for experiments in Group B 71

9 Comparison of results across NN models 77

10 Conclusion 81
10.1 Recommendations 81
10.2 Objectives 82
10.3 Future work 82
10.3.1 Deepening the study 83
10.3.2 Broadening the study 83

Appendix 85

A Appendix **85**

A.1 LSTM model results . . . 85

A.1.1 Test 1 results . . . 85

A.1.2 Test 2 results . . . 88

A.1.3 Test 3 results . . . 92

A.1.4 Test 4 results . . . 95

A.1.5 Test 5 results . . . 98

A.1.6 Test 6 results . . . 101

A.1.7 Test 7 results . . . 104

A.1.8 Test 8 results . . . 107

A.1.9 Test 9 results . . . 110

A.1.10 Test 10 results . . . 113

A.1.11 Test 11 results . . . 116

A.1.12 Test 12 results . . . 119

A.2 CNN-LSTM model results . . . 122

A.2.1 Test 1 results . . . 122

A.2.2 Test 2 results . . . 125

A.2.3 Test 3 results . . . 129

A.2.4 Test 4 results . . . 132

A.2.5 Test 5 results . . . 136

A.2.6 Test 6 results . . . 139

A.2.7 Test 7 results . . . 143

A.2.8 Test 8 results . . . 146

A.2.9 Test 9 results . . . 150

A.2.10 Test 10 results . . . 153

A.2.11 Test 11 results . . . 157

A.2.12 Test 12 results . . . 160

A.3 Multi-head model results . . . 164

A.3.1 Test 1 results . . . 164

A.3.2 Test 2 results . . . 167

A.3.3 Test 3 results . . . 171

A.3.4 Test 4 results . . . 174

A.3.5 Test 5 results . . . 178

A.3.6 Test 6 results . . . 181

A.3.7 Test 7 results . . . 185

A.3.8 Test 8 results 188
A.3.9 Test 9 results 192
A.3.10 Test 10 results 195
A.3.11 Test 11 results 199
A.3.12 Test 12 results 202
A.4 Input features 205

Bibliography **207**

List of Figures

1.1 A schematic representation of when to import and export. 2
1.2 A schematic representation of supervised learning. 4
1.3 A schematic representation of unsupervised learning. 5

2.1 A schematic representation of a radial basis function neural network. 10

3.1 An illustration of a 9 and 21-day moving average. 17
3.2 An illustration of Bollinger bands. 17
3.3 An illustration of a 14-day RSI 18
3.4 An illustration of Heiken-Ashi candlesticks. 19
3.5 An illustration of Ichimoku Kinko Clouds. 21
3.6 An illustration of implied volatility. 21
3.7 A schematic representation of the structure of a typical Convolutional Neural Network. 22
3.8 An illustration of a single RNN cell. 23
3.9 An illustration of how the hidden state is passed to the next RNN cell. 24
3.10 A schematic representation of a LSTM cell as illustrated by Géron [21]. 24
3.11 A schematic representation of the implemented LSTM model. 27
3.12 A simplified schematic representation of the implemented CNN LSTM model. . 28
3.13 A detailed schematic representation of the implemented CNN LSTM model. . . 28
3.14 A simplified schematic representation of the implemented multi-head model. . . 28
3.15 A detailed schematic representation of the implemented multi-head model. . . . 29

4.1 A scree plot of the eigenvalues obtained through PCA. 32

5.1 The predictions made by the Naïve forecast (blue line) compared to the actual values (red line). 37
5.2 An graphical representation of the accuracy and error of the Naïve model. . . . 38

5.3 The predictions made by the SES model (blue line) compared to the actual values (red line). 38
5.4 A graphical representation of the accuracy and error of the SES model. 39
5.5 The predictions made by using MR (blue line) compared to the actual values (red line). 40
5.6 An illustration of which variables had the greatest coefficients in the MR model. 40
5.7 A graphical representation of the accuracy and error of the MR model. 44
5.8 The distribution of the residuals of the MR model. 44
5.9 A scatter plot of the residuals of the MR model. 45

6.1 The best predictions made by the LSTM model for test 1. 48
6.2 An illustration of the change in accuracy and error of the LSTM model for test 1. 48
6.3 A heatmap that illustrates the best accuracy of the LSTM model for test 1. . . 49
6.4 An illustration of different thresholds accuracies for test 1 of the LSTM model. 50
6.5 An illustration of the classification accuracy for test 1 of the LSTM model. . . 51
6.6 The best predictions made by the LSTM model for test 5. 52
6.7 An illustration of the change in accuracy and error of the LSTM model for test 5. 53
6.8 A heatmap that illustrates the best accuracy of the LSTM model for test 5. . . 53
6.9 An illustration of different thresholds accuracies for test 5 of the LSTM model. 54
6.10 An illustration of the classification accuracy for test 5 of the LSTM model. . . 55

7.1 The best predictions made by the CNN-LSTM model for test 1. 58
7.2 An illustration of the change in accuracy and error of the CNN-LSTM model for test 1. 58
7.3 An illustration of test 1 accuracies with different architectures of the CNN-LSTM model. 59
7.4 An illustration of the changing accuracies when varying the accuracy threshold for test 1 of the CNN-LSTM model. 60
7.5 The accuracy obtained when predicting price movement for test 1 of the CNN-lSTM model. 61
7.6 The best predictions made by the LSTm model for test 5. 62
7.7 An illustration of the change in accuracy and error of the CNN-LSTM model for test 5. 62
7.8 An illustration of test 5 accuracies obtained with different architectures of the CNN-LSTM model. 63
7.9 An illustration of the changing accuracies when varying the accuracy threshold for test 5 of the CNN-LSTM model. 64
7.10 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5. 65

8.1 The best predictions made by the Multi-head model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set. 68

8.2 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 7. 68

8.3 A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the Multi-head model. 69

8.4 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025. 70

8.5 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7. 71

8.6 The best predictions made by the Multi-head model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set. 72

8.7 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 11. 72

8.8 A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the Multi-head model. 73

8.9 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025. 74

8.10 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11. 75

A.1 The best predictions made by the LSTM model for test 1 with 1 previous day as input using a walk-forward validation approach on the test set. 86

A.2 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 1. 86

A.3 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 1 with 1 previous day as input. 87

A.4 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025. 87

A.5 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1. 88

A.6 The best predictions made by the LSTM model for test 2 with 1 previous day as input using a walk-forward validation approach on the test set. 89

A.7 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 2. 89

A.8 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 2 with 1 previous day as input. 90

A.9 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025. 90

A.10 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2. 91

A.11 The best predictions made by the LSTM model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set. 92

A.12 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 3. 93

A.13 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 3 with 1 previous day as input. 93

A.14 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025. 94

A.15 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3. 94

A.16 The best predictions made by the LSTM model for test 4 with 1 previous day as input using a walk-forward validation approach on the test set. 95

A.17 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 4. 96

A.18 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 4 with 1 previous day as input. 96

A.19 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025. 97

A.20 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4. 97

A.21 The best predictions made by the LSTM model for test 5 with 1 previous day as input using a walk-forward validation approach on the test set. 98

A.22 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 5. 99

A.23 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 5 with 2 previous day as input. 99

A.24 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025. 100

A.25 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5. 100

A.26 The best predictions made by the LSTM model for test 6 with 1 previous day as input using a walk-forward validation approach on the test set. 101

A.27 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 6. 102

A.28 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 6 with 1 previous day as input. 102

A.29 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025. 103

A.30 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6. 103

A.31 The best predictions made by the LSTM model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set. 104

A.32 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 7. 105

A.33 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 7 with 1 previous day as input. 105

A.34 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025. 106

A.35 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7. 106

A.36 The best predictions made by the LSTM model for test 8 with 1 previous day as input using a walk-forward validation approach on the test set. 107

A.37 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 8. 108

A.38 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 8 with 1 previous day as input. 108

A.39 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025. 109

A.40 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8. 109

A.41 The best predictions made by the LSTM model for test 9 with 1 previous day as input using a walk-forward validation approach on the test set. 110

A.42 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 9. 111

A.43 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 9 with 1 previous day as input. 111

A.44 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025. 112

A.45 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9. 112

A.46 The best predictions made by the LSTM model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set. 113

A.47 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 10. 114

A.48 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 10 with 1 previous day as input. 114

A.49 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025. 115

A.50 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10. 115

A.51 The best predictions made by the LSTM model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set. 116

A.52 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 11. 117

A.53 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 11 with 1 previous day as input. 117

A.54 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025. 118

A.55 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11. 118

A.56 The best predictions made by the LSTM model for test 12 with 1 previous day as input using a walk-forward validation approach on the test set. 119

A.57 A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 12. 120

A.58 A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 12 with 5 previous day as input. 120

A.59 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025. 121

A.60 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12. 121

A.61 The best predictions made by the CNN-LSTM model for test 1 with 2 previous day as input using a walk-forward validation approach on the test set. 122

A.62 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 1. 123

A.63 A graphical representation of the influence that different architectures have on prediction accuracy for test 1 of the CNN-LSTM model. 123

A.64 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025. 124

A.65 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1. 125

A.66 The best predictions made by the CNN-LSTM model for test 2 with 3 previous days as input using a walk-forward validation approach on the test set. 126

A.67 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 2. 126

A.68 A graphical representation of the influence that different architectures have on prediction accuracy for test 2 of the CNN-LSTM model. 127

A.69 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025. 127

A.70 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2. 128

A.71 The best predictions made by the CNN-LSTM model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set. 129

A.72 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 3. 130

A.73 A graphical representation of the influence that different architectures have on prediction accuracy for test 3 of the CNN-LSTM model. 130

A.74 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025. 131

A.75 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3. 132

A.76 The best predictions made by the CNN-LSTM model for test 4 with 2 previous days as input using a walk-forward validation approach on the test set. 133

A.77 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 4. 133

A.78 A graphical representation of the influence that different architectures have on prediction accuracy for test 4 of the CNN-LSTM model. 134

A.79 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025. 134

A.80 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4. 135

A.81 The best predictions made by the CNN-LSTM model for test 5 with 3 previous days as input using a walk-forward validation approach on the test set. 136

A.82 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 5. 137

A.83 A graphical representation of the influence that different architectures have on prediction accuracy for test 5 of the CNN-LSTM model. 137

A.84 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025. 138

A.85 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5. 139

A.86 The best predictions made by the CNN-LSTM model for test 6 with 2 previous days as input using a walk-forward validation approach on the test set. 140

A.87 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 6. 140

A.88 A graphical representation of the influence that different architectures have on prediction accuracy for test 6 of the CNN-LSTM model. 141

A.89 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025. 141

A.90 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6. 142

A.91 The best predictions made by the CNN-LSTM model for test 7 with 3 previous days as input using a walk-forward validation approach on the test set. 143

A.92 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 7. 144

A.93 A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the CNN-LSTM model. 144

A.94 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025. 145

A.95 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7. 146

A.96 The best predictions made by the CNN-LSTM model for test 8 with 3 previous days as input using a walk-forward validation approach on the test set. 147

A.97 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 8. 147

A.98 A graphical representation of the influence that different architectures have on prediction accuracy for test 8 of the CNN-LSTM model. 148

A.99 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025. 148

A.100 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8. 149

A.101 The best predictions made by the CNN-LSTM model for test 9 with 4 previous days as input using a walk-forward validation approach on the test set. 150

A.102 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 9. 151

A.103 A graphical representation of the influence that different architectures have on prediction accuracy for test 9 of the CNN-LSTM model. 151

A.104 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025. 152

A.105 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9. 153

A.106 The best predictions made by the CNN-LSTM model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set. 154

A.107 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 10. 154

A.108 A graphical representation of the influence that different architectures have on prediction accuracy for test 10 of the CNN-LSTM model. 155

A.109 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025. 155

A.110 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10. 156

A.111 The best predictions made by the CNN-LSTM model for test 11 with 3 previous days as input using a walk-forward validation approach on the test set. 157

A.112 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 11. 158

A.113 A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the CNN-LSTM model. 158

A.114 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025. 159

A.115 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11. 160

A.116 The best predictions made by the CNN-LSTM model for test 12 with 5 previous days as input using a walk-forward validation approach on the test set. 161

A.117 A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 12. 161

A.118 A graphical representation of the influence that different architectures have on prediction accuracy for test 12 of the CNN-LSTM model. 162

A.119 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025. 162

A.120 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12. 163

A.121 The best predictions made by the Multi-head model for test 1 with 1 previous day as input using a walk-forward validation approach on the test set. 164

A.122 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 1. 165

A.123 A graphical representation of the influence that different architectures have on prediction accuracy for test 1 of the Multi-head model. 165

A.124 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025. 166

A.125 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1. 167

A.126 The best predictions made by the Multi-head model for test 2 with 1 previous day as input using a walk-forward validation approach on the test set. 168

A.127 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 2. 168

A.128 A graphical representation of the influence that different architectures have on prediction accuracy for test 2 of the Multi-head model. 169

A.129 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025. 169

A.130 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2. 170

A.131 The best predictions made by the Multi-head model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set. 171

A.132 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 3. 172

A.133 A graphical representation of the influence that different architectures have on prediction accuracy for test 3 of the Multi-head model. 172

A.134 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025. 173

A.135 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3. 174

A.136 The best predictions made by the Multi-head model for test 4 with 1 previous day as input using a walk-forward validation approach on the test set. 175

A.137 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 4. 175

A.138 A graphical representation of the influence that different architectures have on prediction accuracy for test 4 of the Multi-head model. 176

A.139 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025. 176

A.140 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4. 177

A.141 The best predictions made by the Multi-head model for test 5 with 1 previous day as input using a walk-forward validation approach on the test set. 178

A.142 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 5. 179

A.143 A graphical representation of the influence that different architectures have on prediction accuracy for test 5 of the Multi-head model. 179

A.144 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025. 180

A.145 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5. 181

A.146 The best predictions made by the Multi-head model for test 6 with 1 previous day as input using a walk-forward validation approach on the test set. 182

A.147 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 6. 182

A.148 A graphical representation of the influence that different architectures have on prediction accuracy for test 6 of the Multi-head model. 183

A.149 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025. 183

A.150 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6. 184

A.151 The best predictions made by the Multi-head model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set. 185

A.152 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 7. 186

A.153 A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the Multi-head model. 186

A.154 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025. 187

A.155 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7. 188

A.156 The best predictions made by the Multi-head model for test 8 with 1 previous day as input using a walk-forward validation approach on the test set. 189

A.157 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 8. 189

A.158 A graphical representation of the influence that different architectures have on prediction accuracy for test 8 of the Multi-head model. 190

A.159 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025. 190

A.160 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8. 191

A.161 The best predictions made by the Multi-head model for test 9 with 1 previous day as input using a walk-forward validation approach on the test set. 192

A.162 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 9. 193

A.163 A graphical representation of the influence that different architectures have on prediction accuracy for test 9 of the Multi-head model. 193

A.164 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025. 194

A.165 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9. 195

A.166 The best predictions made by the Multi-head model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set. 196

A.167 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 10. 196

A.168 A graphical representation of the influence that different architectures have on prediction accuracy for test 10 of the Multi-head model. 197

A.169 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025. 197

A.170 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10. 198

A.171 The best predictions made by the Multi-head model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set. 199

A.172 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 11. 200

A.173 A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the Multi-head model. 200

A.174 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025. 201

A.175 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11. 202

A.176 The best predictions made by the Multi-head model for test 12 with 5 previous days as input using a walk-forward validation approach on the test set. 203

A.177 A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 12. 203

A.178 A graphical representation of the influence that different architectures have on prediction accuracy for test 12 of the Multi-head model. 204

A.179 An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025. 204

A.180 An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12. 205

List of Tables

1.1 A summary of both scenarios of a farmer importing and exporting goods at different exchange rates. 2

3.1 An example of restructuring time series data into a labelled dataset. 25

3.2 A table showing the input features for all the different tests that were conducted. 26

4.1 A table showing the input features for all the different tests that were conducted. 33

5.1 A description of all variables used in the MR model. 41

5.2 The p-values obtained for all the variables identified by PCA. The result of the null hypothesis, H_0, is shown in last column. 42

5.3 The different ANOVA and performance measures obtained when the variables that failed the hypothesis test were removed from the MR model. 43

6.1 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1. 50

6.2 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1. 51

6.3 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5. 54

6.4 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5. 55

7.1 The changing accuracies when varying the accuracy threshold for test 1 of the CNN-LSTM model 60

7.2 The accuracy obtained when predicting price movement for test 1 of the CNN-lSTM model. 60

7.3 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5. 64

7.4 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5. 64

8.1 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7. 70

8.2 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7. 70

8.3 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11. 74

8.4 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11. 74

9.1 The accuracy, in percentage, obtained by all the models for all the tests conducted. 77

9.2 The MSE obtained by all the models for all the tests conducted. 78

9.3 The accuracy obtained by the different models for test 1 and test 5 with varying accuracy thresholds. 79

9.4 The accuracy obtained by the different models for test 7 and test 11 with varying accuracy thresholds. 79

9.5 The accuracy obtained by all the models for all the tests conducted when the direction of price movement is considered. 80

9.6 The overall best performing tests for each NN. 80

A.1 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1. 85

A.2 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1. 86

A.3 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2. 89

A.4 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2. 91

A.5 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3. 92

A.6 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3. 93

A.7 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4. 95

A.8 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4. 96

A.9 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5. 98

A.10 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5. 99

A.11 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6. 101

A.12 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6. 102

A.13 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7. 104

A.14 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7. 105

A.15 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8. 107

A.16 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8. 108

A.17 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9. 110

A.18 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9. 111

A.19 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10. 113

A.20 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10. 114

A.21 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11. 116

A.22 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11. 117

A.23 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12. 119

A.24 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12. 120

A.25 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1. 124

A.26 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1. 124

A.27 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2. 128

A.28 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2. 128

A.29 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3. 131

A.30 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3. 131

A.31 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4. 135

A.32 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4. 135

A.33 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5. 138

A.34 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5. 138

A.35 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6. 142

A.36 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6. 142

A.37 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7. 145

A.38 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7. 145

A.39 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8. 149

A.40 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8. 149

A.41 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9. 152

A.42 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9. 152

A.43 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10. 156

A.44 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10. 156

A.45 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11. 159

A.46 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11. 159

A.47 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12. 163

A.48 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12. 163

A.49 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1. 166

A.50 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1. 166

A.51 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2. 170

A.52 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2. 170

A.53 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3. 173

A.54 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3. 173

A.55 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4. 177

A.56 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4. 177

A.57 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5. 180

A.58 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5. 180

A.59 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6. 184

A.60 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6. 184

A.61 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7. 187

A.62 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7. 187

A.63 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8. 191

A.64 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8. 191

A.65 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9. 194

A.66 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9. 194

A.67 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10. 198

A.68 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10. 198

A.69 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11. 201

A.70 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11. 201

A.71 The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12. 205

A.72 The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12. 205

A.73 A table showing the input features for all the different tests that were conducted. 206

CHAPTER 1

Introduction

The development of a country is fuelled by several important factors and industries. One such factor is industrialisation. In 1760 when the industrial revolution started, countries who adopted the new manufacturing procedures earlier than others were sling-shotted forward in terms of economic power and per capita income. Generally, the earlier a country initialises new technologies and ideas, the faster it develops. Although the onset of industrialisation was a slow process, Gollin *et al.* [23] state that an improvement in agricultural productivity can shorten the onset of industrialisation and that "growth in agricultural productivity is central to development".

Agriculture in South Africa is a major industry which contributed around 10% of the total export earnings, at a value of US$11.1 billion, in the 2018 financial year [7]. This contribution could fluctuate according to several factors; such as whether the specific commodity is in demand or not, a drought causing low yield or the strength/weakness of the South African Rand (ZAR) versus other international currencies. The latter is a crucial factor that requires constant monitoring — a farmer who relies heavily on imports/exports is directly impacted by any fluctuation in the exchange rate.

Consider a wine farmer in the Stellenbosch Winelands who exports wine and imports agricultural equipment from the United States of America (USA). When the farmer's wine is exported, the farmer will be paid in American dollars (USD) and when s/he import equipment, they will pay in ZAR. The influence of the exchange rate becomes apparent when the farmer in the aforementioned transaction wishes to convert the foreign currency into local currency. Deciding when to do this conversion can be immensely challenging when the volatile exchange rate between USD and ZAR is considered. A weaker ZAR will lead to an increase in purchase cost (loss of funds) when importing, but an increase in ZAR received when exporting.

Consider Figure 1.1 where the farmer decides to import equipment worth US$10 000 at ZAR13 per USD, and export wine worth US$20 000 at ZAR15 per USD. The farmer therefore pays ZAR130 000 for the equipment and receives ZAR300 000 for their wine. However, if the opposite scenario (both scenarios summarised in Table 1.1) occurred where the equipment is imported at ZAR15 per USD and the wine is exported at ZAR13 per USD, the farmer will pay ZAR150 000 for the equipment, a increase of ZAR20 000, and receive ZAR260 000 for their wine, a decrease of ZAR40 000.

Although the decrease of ZAR20 000/ZAR40 000 might seem small, if the funds received/paid is scaled to millions of USD or ZAR, the increase/decrease of funds could lead to a major loss or profit. Imagine the same scenario with the farmer but the equipment is worth US$10 million and the wine is worth US$1 million. In the latter example where the equipment is imported

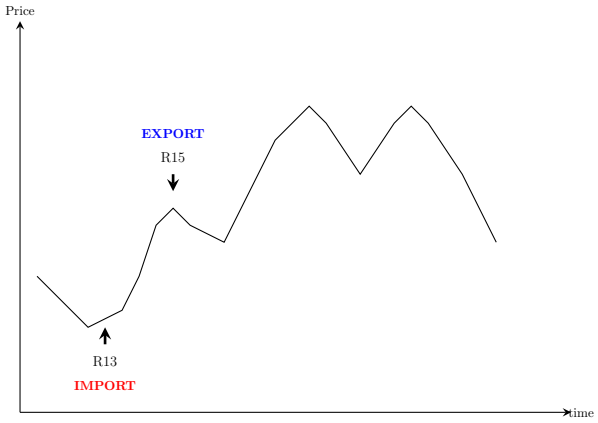


FIGURE 1.1: *A schematic representation indicating when to import and export according to the strength or weakness of the exchange rate.*

	Worth ($)	Type	Exchange rate (USD/ZAR)	Fund type	Total funds (R)
Equipment	10 000	import	13.000	Paid	130 000
Equipment	10 000	import	15.000	Paid	150 000
Wine	20 000	export	15.000	Received	300 000
Wine	20 000	export	13.000	Received	260 000

TABLE 1.1: *A summary of both scenarios of a farmer importing and exporting goods at different exchange rates.*

at ZAR15 per USD and the wine is exported at ZAR13 per USD; the farmer will pay ZAR150 million for the equipment, an increase of ZAR20 million, and will receive ZAR13 million, a decrease of ZAR2 million. The ability to predict/anticipate the movement of the exchange rate in the near future could therefore help farmers make informed business decisions and potentially save them millions.

1.1 Foreign exchange

Foreign Exchange, as defined by Chen [16], "is the trading of one currency for another". These trades or transactions occur in the foreign exchange market. Exchanging local currency to a foreign currency for an arbitrary reason at the bank is not the same as trading in the foreign exchange market. For the purpose of this study, trading in the foreign exchange market is not considered, but rather the scenario of going to the bank (assuming all banks offer the most recent exchange rate) and exchanging local currency to a foreign currency. Scheduling a visit to the bank (when the exchange rate is in your favour) is therefore a difficult task, but knowing and understanding the data that influences these rates could assist in timing such a visit.

Data has become a form of currency that can govern both the success and failure of almost every business or individual. Raw, unprocessed data and the type of data that is utilised, when the financial markets are considered, form the basis for new discoveries. According to Prado [39], financial data has four main types: fundamental, market, analytics and alternative data.

Fundamental data, as stated by Prado [39], "is extremely regularised and low frequency" as it predominately contains quarterly reported account data. Several examples for a country include:

GDP (Gross Domestic Product), CPI (Consumer Price Index), unemployment rate, and interest rate; and for business include: assets, liabilities, sales, and costs/earnings. This data is easily accessible and available to the public, making new insights or discoveries highly unlikely.

In comparison to sparse fundamental data, market data generates over 10 terabytes (TB) of data every day [39]. The sheer volume of data can be difficult to summarise, but Boginski *et al.* [11] states that it "is usually visualised by thousands of plots" and consists of any trading activity that takes place at an exchange. Once access to market data is obtained, unique strategies can be applied to gain insight and potentially make new discoveries. Several examples of market data include: price, yield, volume, open interest, quotes/cancellations and implied volatility.

Analytic data on the other hand is not restricted to a specific data type and could be a combination of the four data types — it could, however, be seen as derivative data where the price is dependent on the data itself [39]. Unlike fundamental and market data, analytic data is only available through investment banks and research firms, who require payment for the data and process the data for clients. This means that a client will obtain preprocessed data, which could have a bias introduced by the bank/firm, and data which is accessible by anyone willing to purchase it. Several data types that could be included in analytic data are analyst recommendations, credit ratings, sentiment analysis and earnings expectations.

Finally, alternative data is acquired by collecting data produced either directly or indirectly by individuals, business processes or sensors [39]. Collecting this data can be controversial as it could include a wide variety of sensitive information that the individual or business may or may not be aware of. Collected data could include some of the following: social media activity or web searches of an individual, transactions, policy changes, and corporate data of a business and weather, locations, CCTV footage, or satellite images from various sensors [39]. A business or individual could rebuff the use of alternative data as the difficultly to process and/or store it could cause major delays to data preprocessing pipelines. This makes alternative data efficacious and unique as not all businesses or individuals have the resources to store and/or process the data, let alone ensure that it is done correctly.

Once the relevant datasets have been obtained or granted access to, data analysis and forecasting models can be applied to gain further insights. However, since financial data is non-stationary and filled with noise, careful consideration is required when selecting forecasting models [47].

1.2 Machine learning

Over the years, Machine learning (ML) has grown from just a concept to a leading analysis/prediction tool used in almost every industry in the world. The immense volume of data generated, mined and collected is the fuel that keeps the interest and development of ML alive. Without data, ML would not be able to perform and advance in the way that it has. However, a wealth of data does not imply that all data is relevant and/or important. Selecting input variables for ML models is vitally important, and, as Xue *et al.* [47] states, "even the best machine learning technique can only learn from an input if there is actually some correlation between input and output variable".

ML is defined by Kirk [29] as "a collection of algorithms, techniques, and tricks of the trade that allow machines to learn from data". However, Géron [21] describes ML as "the science (and art) of programming computers so they can learn from data". Given the black box nature of ML algorithms, describing them as "art" seems more appropriate as it highlights the skill required to build and utilise them (anyone can pick up a paintbrush but not everyone can paint

the Mona Lisa). It is a process where machines learn to understand and extract information from data in the same way humans do, if not better. Similarly to humans who learn in different ways, the "learning" in ML falls into four categories: supervised, unsupervised, semi-supervised and reinforcement learning [29].

Supervised learning tries to solve the problem of fitting a function or finding a function approximation on a specific dataset [29]. It tries to achieve the "best fit" on data by trying to map an input, $\boldsymbol{x}$, to an output, $\boldsymbol{y}$ (similar to Figure 1.2 where a best fit line is fitted to a dataset). The end goal is to have a mapping function that can accurately predict the output, $\boldsymbol{y}$, given a new unseen input, $\boldsymbol{x}$ [12]. The algorithm functions on one primary factor, having labelled data (the desired solution). This means that for all samples of data, there exists a label (defining what the sample represents) associated with each sample. Consider the question "Can hair colour and length be used to predict an individual's eye colour". To answer the question the hair colour, hair length, and eye colour of each individual of a business could be recorded (as seen in Table 1.2). The eye colour of each individual would then be the label for each sample of data.

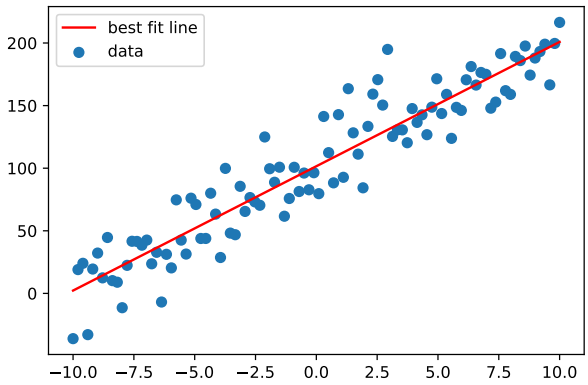


FIGURE 1.2: *A schematic representation of how a supervised learning algorithm would fit a function to a set of data.*

Hair Colour	Hair Length (cm)	Eye Colour
blonde	30	blue
brunette	50	green
black	30	brown

TABLE 1.2: *An example of what the recorded data could look like if the hair colour, hair length and eye colour of three individuals at a business were recorded.*

When data lacks any labelled data points, unsupervised learning can be used to identify the underlying structure, distribution and correlation of the data [12]. Unlike supervised learning, unsupervised learning has no distinction between right and wrong — the algorithm is left to make decisions and extract the necessary and relevant information by itself. Think of a supermarket that would like to redesign the layout of a store, but do not know which items should be placed next/close to one another. An unsupervised learning algorithm could then be used to detect patterns in customer purchasing patterns. Figure 1.3 illustrates this by showing that a customer who purchases large quantities of sweets and chips, generally purchase paper cups too (blue cluster). This could mean that a customer is purchasing supplies for a birthday party

(paper cups are also purchased) and it could be beneficial to place all three items next/close to one another on the shelf.

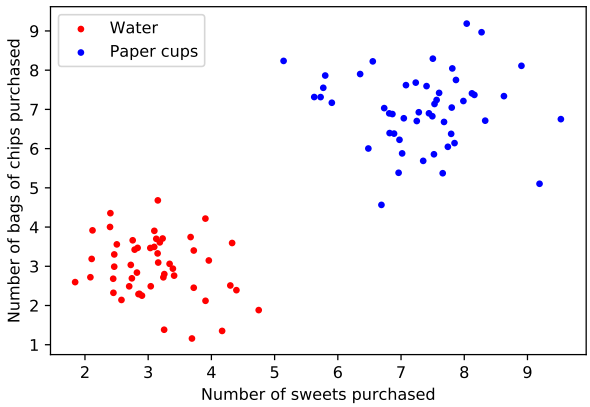


FIGURE 1.3: *A schematic representation of how an unsupervised learning algorithm could cluster customer data according to what items were purchased most frequently.*

In the event that only a percentage of samples in the data have labels, semi-supervised learning can be used. As the name suggests, an algorithm is used that utilizes aspects from both supervised and unsupervised learning. Consider an example from Géron [21] where family photos are uploaded to a cloud-based platform that identifies the individual(s) in the photos. This process utilises unsupervised learning to cluster photos of an individual together, identifying the specific photos an individual is in, and then uses supervised learning, with as little as one labelled photo of each individual, to predict the name of the individual.

Lastly, reinforcement learning differs substantially from the previous three learning methods. It involves a learning system, known as an agent, that can observe its surroundings and choose what actions to perform [21]. In the event that an action leads to an undesired result, the agent is given a penalty, whereas if the desired result is obtained, a reward is given. The algorithm must learn what the best strategy, called a policy, is to obtain the most rewards in a given time period. Imagine a robot that can move in any direction. If the robot chooses a direction and does not hit anything, a reward is given in the form of points. However, if the robot decides to move in a direction resulting in a collision with an object, a penalty will be given and points will be deducted. The robot would then be placed in a room with random objects placed on the floor and would be required to manoeuvre around the room. As the robot moves around, colliding into objects or avoiding them, it learns which path leads to the maximum number of attainable points.

The different types of learning algorithms that can be utilised in ML therefore gives ML algorithms the ability to be applied in any industry. One such field of application is financial markets. Since financial data is inherently non-linear, it would be appropriate to try and apply ML techniques to gain further insight into the financial markets. One ML technique that McNelis [32] states could "offer a powerful alternative to linear models for forecasting, classification, and risk assessment in finance and economics", is neural networks (NNs).

According to, Osinga [38], "neural networks are remarkably good at finding patterns in data". It was therefore inevitable that NNs would start playing an ever growing role in analysis and making predictions in the finance industry.

1.3 Problem description

The aim of this study is to investigate the importance of selecting input features and their effect on NN performance when predicting the daily closing price of the USD/ZAR exchange rate. Three different NNs will be implemented and tested with 12 different input features. Each NN and the respective input feature for each test will be tested on different NN architectures. This ensures that each input feature is given the opportunity to provide the best results for each NN.

1.4 Thesis scope

The scope of this study will be limited to the following:

1. Only the architecture of the three NNs will be changed throughout testing. All other hyper parameters will either be chosen and kept constant or remain at their default values.
2. Only two types of NNs will be considered — the Convolutional Neural Network and Long short-term memory network.
3. Features that are used for testing will be limited to those suggested by a professional foreign exchange trader.
4. Training and testing of the implemented NNs will be done on one computer alone. Code optimisation will not be considered.

1.5 Thesis objectives

For the purpose of this study, the following objectives are investigated:

1. Provide a background to finance and the importance of knowing what the exchange rate might potentially do;
2. Provide a background to machine learning and the applications in finance;
3. Describe the data and models that were used and implemented;
4. Describe the experimental setup;
5. Investigate the forecasting performance of standard statistical methods;
6. Compare results between the different NNs as well as how the input features effected those results;
7. Discuss potential directions that could be taken for further research.

1.6 Thesis layout

In Chapter 2 the relevant literature will be discussed. A detailed description of the literature that was reviewed will be given as well as a description of which literature will be used in this study.

The data collected as well as the models that were used will be described in Chapter 3. A brief description of the data will be given, including the input features for the NNs. The NN models will also be described and shown graphically.

In Chapter 4 Principal Component Analysis and the findings thereof will be discussed. The statistical forecasting models that were implemented as well as their results will be described in Chapter 5.

The results for the three NN models will be discussed in Chapters 6, 7 and 8, with a final comparison of the results described in Chapter 9.

Finally, Chapter 10 provides the conclusion of this study which contains recommendations based off the obtained results and a description of possible future work that can be done.

In the next chapter a summary of the reviewed literature will be given.

CHAPTER 2

Literature review

Contents

2.1 Foreign exchange 9
2.2 Stock market 11
2.3 Conclusion 13

ML techniques and the use of NNs have seen a huge surge in both research and use in practice. Finding literature that is relevant to the scope within this wealth of information is the guide for this chapter. Relevant literature would require the use of neural networks, or other ML techniques, to build regression/classification models in the financial industry, but specifically for foreign exchange trading. Considering both regression and very specific classification applications paints a broader understanding of how ML is used and can be applied. The remainder of this chapter will discuss literature in which ML techniques were applied to both foreign exchange and the stock market.

2.1 Foreign exchange

Feature selection has the potential to influence model performance, but more importantly it could provide a glimpse into the proverbial black box that is ML [39]. By experimenting with different features and analysing the impact they have on performance, an understanding of how ML models identify important information could be attained.

Cao and Tay [14] proposed a method of using saliency[1] analysis (SA) on support vector machines (SVMs) to determine important features on five real futures contracts: Standard&Poor 500 stock index futures, United States 30-year government bond, United States 10-year government bond, German 10-year government bond and French government stock index futures. The dataset, with a total of 17 features, was constructed from transforming the daily closing price into 3 lagged closing prices (subtracting a 15-day exponential moving average from the closing price), and 14 lagged relative difference in percentage of price (RDP) values. The performance gain from using SA on the set of features saw the biggest decrease in normalised mean square error (NMSE) of 0.397 when using the German 10-year government bond.

[1]Salience is defined by Stevenson [45] as "the quality of being particularly noticeable or important".

SVMs are a specific non-parametric ML algorithm (they do not make any assumptions about the mapping function) which is most commonly used in classification problems [12]. However, providing a SVM with continuous data allows the algorithm to adapt and behave like a regression model [10]. This adaptation is done by fitting the error within a specified threshold, unlike regular regression which tries to minimise the error.

Cao and Tay [15] expanded on their previous study in which a SVM with adaptive parameters, a multi-layer neural network with back-propagation (BPNN) and the regularised radial basis function neural network (RBF) were compared. The dataset used was similar to the one used in their earlier study (same five real futures contracts), but with the main difference being the input features. The new features were constructed from four lagged RDP values and one lagged closing price obtained by subtracting a 100-day exponential moving average (EMA100) from the closing price. Cao and Tay [15] found that the SVM had superior performance when compared to the benchmark of the BPNN, but had similar performance when compared to the benchmark of the RBF.

A RBF is a type of artificial neural network (ANN) constructed from n basis functions, shown as $h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})$ in Figure 2.1. These functions produce the network output, represented as $f(\boldsymbol{x})$ in Figure 2.1. Radial functions, as stated by Orr [37], are unique in that "their response decreases (or increase) monotonically with distance from a central point".

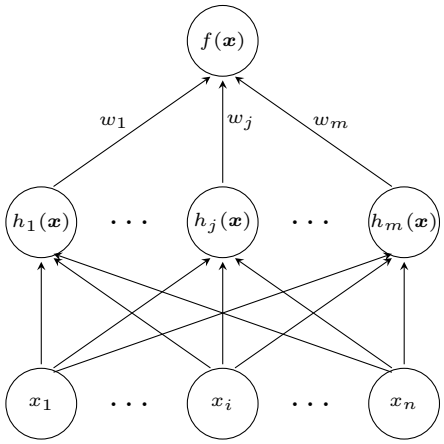


FIGURE 2.1: *A schematic representation from Orr [37] showing the network architecture for a radial basis function neural network.*

Although the task of selecting input features can be troublesome, Cao and Tay [15] and Cao and Tay [14] show that by using different models, a method to select input features that results in higher performance can be achieved. However, another approach to measure the impact on performance of input features is to measure the difference in predictive power when using different input features with different models.

Hussain *et al.* [27] proposed a Dynamic Ridge Polynomial Neural Network (DRPNN) to predict the daily closing price of two foreign exchange rates. The dataset they used matched that used by Cao and Tay [15], but with a 15 day exponential moving average (EMA15) instead of the EMA100. The addition of the EMA15 was to ensure that any underlying structure or information within the original series was retained, as RDP transformations could remove information from the series [15]. The predicted output from the neural network was then modified to be a RDP for five days ahead. The proposed DRPNN differs from a feedforward Ridge Polynomial Neural Network (RPNN) with one recurrent link from the output to input layer, giving the neural network an internal memory. During the learning process, additional Pi-Sigma Neural network

(PSNN) units with higher orders will be added until either the desired mean square error (MSE) or the number of epochs has been reached. Hussain *et al.* [27] then compared the proposed DRPNN to three benchmark models: the multilayer perceptron (MLP), PSNN and RPNN. They found that the DRPNN had the highest annualised return (the measure of model performance for their study) of 88.74% for the one exchange rate (an increase of 0.43% from the RPNN which performed second best), and a 86.21% annualised return for the second exchange rate (an increase of 0.57% from the RPNN and MLP). Although the increase in annualised return is minimal when using the DRPNN, the number of epochs required for training decreased by a maximum of 99.8333%. This decrease leads to faster convergence as well as a decline in computational time required to train the model.

Ni *et al.* [35] proposed a hybrid Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) model, referred to as C-RNN, to predict the closing price of nine currency pairs. The dataset used for training and predictions ranged from June 2008–May 2018 and consisted of four features: the open, high, low and closing price. As CNNs are generally used in classification applications, such as image recognition, they are well known to capture spatial relationships within data. However, as foreign exchange data has an underlying temporal structure, the RNN is required to capture any temporal characteristics. By combining a CNN and RNN, the C-RNN proposed by Ni *et al.* [35] can therefore utilise any and all spatio-temporal characteristics within foreign exchange data. They used a Root Mean Square Error (RMSE) to evaluate performance and compared the results obtained by the C-RNN to a CNN and Long Short-Term Memory (LSTM) network. They found that the C-RNN outperformed both the CNN and LSTM network on all nine currency pairs.

The reviewed literature in this section provided a foundation on which this study was built. These studies showed that a lot of experimentation is required to obtain relevant results, and that input features effect every model differently. However, these studies were all focused on data in the foreign exchange market. Studies conducted within the stock market were also considered to gain further insight into possible relationships between input features and predictive power.

2.2 Stock market

As with all industries in the world, several subdivisions exist within the financial industry. Researchers using ML techniques would therefore follow the logical progression and branch out into different subdivisions, shifting their focus from the foreign exchange market to the stock market. Financial market analysis and assumptions such as Mean Reversion (MRV) and Moving Average Reversion (MAR) have shown that the stock market is predictable, unlike the foreign exchange market [49]. MRV is the assumption that the stock price will eventually tend to the historical average of the stock price [17]. MAR is an extension of MRV where the stock price is assumed to tend towards a specified moving average of the historical stock price [31]. Research in the field of applying ML techniques to the stock market increased substantially over the years. However, key financial areas such as portfolio optimisation, stock market prediction, financial information processing and trade execution strategies have received the most attention [49].

Hiransha *et al.* [25] chose to implement four different neural network architectures and compare their predictive capabilities when predicting the daily closing stock price of three companies in the National Stock Exchange (NSE) of India and two companies from the New York Stock Exchange (NYSE). The implemented neural network architectures consisted of a MLP, RNN, LSTM network and CNN. The training dataset was constructed with one feature, the daily closing price of TATAMOTORS from the NSE of India. The models were then tested on two

different test sets (constructed from only the daily closing stock price). The first test set consisted of three companies from the NSE of India and the second derived from the top two active stocks in the NYSE. The justification given to choose two different stock exchanges when training and testing the models was that the models would be able to learn the dynamics between different stock exchanges. Hiransha *et al.* [25] found that the CNN produced the most accurate results, outperforming the other neural networks and a benchmark ARIMA model, whilst being able to capture underlying trends in the movement of the stock price. They suggested that a hybrid network should be investigated next to expand on the current research.

Zhang *et al.* [49] predict the daily closing price of the Standard & Poor's 500 (S&P 500) Index, Shanghai Composite Index in China, Internation Business Machine (IBM) share price, Microsoft Corporation (MSFT) share price, and Ping An Insurance Company of China (PAICC) share price by using a Generative Adversarial Network (GAN) architecture. The constructed dataset used in their study consisted of 7 features, namely: Open, High, Low and Closing Prices, Volume, Turnover Rate and a 5-day moving average of the closing price. The proposed GAN framework trains two different models, a generator and discriminator. The generator is used to generate fake data which the discriminator then has to distinguish from the real data. The fake data is derived from adding the prediction of all 7 features at time $t+1$, $\hat{\boldsymbol{x}}_{t+1}$, of the LSTM generator to the real values of the 7 features at time t, forming a vector

$$\boldsymbol{x}_{fake} = [\boldsymbol{x}_t, \ldots, \hat{\boldsymbol{x}}_{t+1}].$$

This data is then given to the discriminator in combination with the real values of all the features at time t and $t+1$. If a point of equilibrium is established, where the discriminator is unable to identify the fake data from the real data, the generator captures the distribution of the fake data. Zhang *et al.* [49] compared the proposed GAN architecture to a LSTM, ANN, and Suppport Vector Regression (SVR) model and found that the GAN outperformed the other models with the greatest decrease in RMSE of 1.3105 and lowest overall Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) values.

Sezer *et al.* [43] propose a stock trading system by combining a genetic algorithm (GA) and deep MLP to determine the best entry and exit points for trades. The constructed dataset is comprised of two technical indicators: Relative Strength Index (RSI) values, used to determine the strength or weakness of a stock price, and a Simple Moving Average (SMA), used to determine the long term trend of the stock price. The RSI values were used to create chromosomes with 8 genes, which was then used by the GA to determine the fittest RSI chromosome. This and the addition of a trend direction would then be used as an input feature for the deep MLP which predicts whether to buy, sell or hold the stock. The model was then tested on several DOW 30 stocks where Sezer *et al.* [43] found that the proposed GA and MLP model outperformed several other trading strategies as well as the buy and hold strategy.

Dash and Dash [19] proposed a trading decision support system that combines a computationally efficient functional link artificial neural network (CEFLANN) and a set of rules based off technical analysis. The constructed dataset consisted of 6 technical indicators: 15-day Moving Average (MA), 26-day Moving Average Convergence and Divergence (MACD), Stochastic KD, RSI and Larry William's R% (WR). All six technical indicators represent different trends, volatility, momentum, and rate of price movements within a given stock or exchange rate. The choice of technical indicators covers all the possible movements of the stock price which allows informative decisions to be made. The proposed trading decision support system starts by calculating all the technical indicators, performing a trend analysis with those indicators and then generating a trading signal. This signal, combined with the technical indicators, are then given as an input to the CEFLANN. The predicted trading signal from the neural network is

then used to classify the trend after which three possible trading decisions is made: buy, sell or hold. Instead of trying to predict the closing price of a stock, Dash and Dash [19] modelled the problem as a classification problem. The performance (profit percentage during a given time) of CEFLANN is then compared to 4 benchmark classification models: the SVM, Naïve Bayesian, K Nearest Neighbour (KNN) and Decision Tree (DT). Dash and Dash [19] found that CEFLANN outperformed the benchmark classifiers on both test datasets (BSE SENSEX and S&P 500 stock indices) with a 4.874% and 2.1204% increase over the Naïve Bayesian classifier, which performed second best on both test data sets.

Bao *et al.* [8] propose a deep learning framework which combines wavelet transforms (WTs), stacked autoencoders (SAEs) and a LSTM to predict the next daily closing price of six stock indices. The entire framework is comprised of three building blocks. The first step is to remove the noise from the raw time series by using WTs, specifically the Haar wavelet. The second step is to extract the features from the time series, using unsupervised learning, with the SAEs. Finally the LSTM predicts the next daily closing stock price by using supervised learning. The use of WT is justified as it can analyse irregular and non-stationary financial time series. However, using the Haar function allows the decomposition of the time series into both time and frequency domains, as well as decrease the time required to process the data. One major advantage of using WT is that the transformation can be applied several times, decreasing the probability of overfitting with each application. Bao *et al.* [8] found that their deep learning framework outperformed a RNN and LSTM network as well as the buy-and-hold strategy. The framework achieved the lowest MAPE value of 0.011 with the second lowest achieving a MAPE value of 0.014. They also performed a profitability test in which their framework outperformed all other models with the highest average annual percentage return for all six stock indices.

2.3 Conclusion

From the literature discussed above, two research papers had the most relevant information and guidance for this study — Ni *et al.* [35] and Hiransha *et al.* [25]. These two research papers are connected as Ni *et al.* [35] found that a hybrid C-RNN model outperformed other NNs, and Hiransha *et al.* [25] suggested that a hybrid model be investigated in further studies. Although both papers are in different financial markets, the expectation is that combining several factors from both papers should lead to models that can be applied to the foreign exchange market.

CHAPTER 3

Data and modelling

Contents

3.1 Features . . . 15
3.1.1 Moving average . . . *16*
3.1.2 Bollinger bands . . . *16*
3.1.3 Relative strength index . . . *18*
3.1.4 Heiken-Ashi candlesticks . . . *19*
3.1.5 Ichimoku kinki cloud . . . *20*
3.1.6 Implied volatility and risk reversal . . . *20*
3.2 Convolutional neural network . . . 22
3.3 Long-short term memory . . . 23
3.4 Data processing . . . 25
3.5 Modelling . . . 26
3.5.1 LSTM network . . . *27*
3.5.2 CNN-LSTM network . . . *28*
3.5.3 Multi-head CNN-LSTM network . . . *28*

There is an almost unlimited choice when it comes to choosing the type of data and features for a model, and thus selecting appropriate features can be difficult. However, with the help from industry experts and research, an informed selection of features can be made. The remainder of this chapter describes all the features and models used in the remainder of this study.

3.1 Features

Features are the building blocks when developing a predictive model. They are independent variables that are given as input to a machine learning algorithm [9]. For the purpose of this study, the daily Open, High, Low and Closing prices are regarded as the baseline features after which technical indicators are included. The use of technical indicators was advised from industry professionals as well as from literature. Colby [18] states that there are 24 advantages to using technical indicators. The main advantages from this list are:

1. They provide a means of quantifying information within the market in an easily digestible manner.

2. A different technical indicator can be applied for the three possible trend directions (up, down, and sideways).

3. They provide a method of removing subjective opinions and allow objective informative decisions to be made. If the technical indicator provides a trading signal, a user can easily decide whether to use the signal or not.

4. They can be updated and tweaked in any way a user seems fit. If an indicator uses a 5-day moving average and a user would like to change it to a 20-day moving average, it is as easy as opening the settings of the indicator and changing the 5 to a 20.

5. They are incredibly easy to access and implement (all trading platforms come with the default indicators).

Consulting with industry professionals and literature lead to the following technical indicators being chosen for this study: Moving averages, Bollinger bands, RSI, Heiken-Ashi candlesticks, Ichimoku kinko cloud, and implied volatility and risk reversals.

3.1.1 Moving average

A moving average (MA), also known as a rolling mean, is a simple method to smooth price fluctuations and gain a better understanding of the underlying trend within the data. An n-day moving average can be specified by a user and used in any financial market. Multiple moving averages are commonly combined to form a strategy. Typically, if two moving averages were used, the one would have a smaller n-day period (referred to as a "fast" moving average) whilst the other would have a longer n day period (referred to as a "slow" moving average). This would then be combined to form a crossover strategy which states that the price would increase if the faster moving average crosses above the slower moving average and vice versa for a decrease in price. Figure 3.1 illustrates what a crossover strategy would look like when applied to the USD/ZAR exchange rate.

3.1.2 Bollinger bands

Bollinger bands are also a popular technical indicator. Bollinger bands use a "support" and "resistance" line two standard deviations away from a n-day MA [18]. The default choice is a 20-day MA as it can be used to describe both the short and long-term trend. What makes Bollinger bands so useful and popular is that it provides a visual representation of how volatile the market is. If the "support" and "resistance" lines start moving further away from one another, it indicates a volatile period in time and the price could change more than expected; whereas if the "support" and "resistance" lines move closer to one another, it indicates a period of "calmness" where the price would change as expected. A typical strategy that could be used to generate buy and sell signals is to look at what band the price touches. If the price touches the upper "resistance" line, it could indicate a possible retracement and a decreasing price. The opposite is true if the price touches the lower "support" line. In Figure 3.2 the price touches and breaks through the bottom support line and immediately pulls back above the support line.

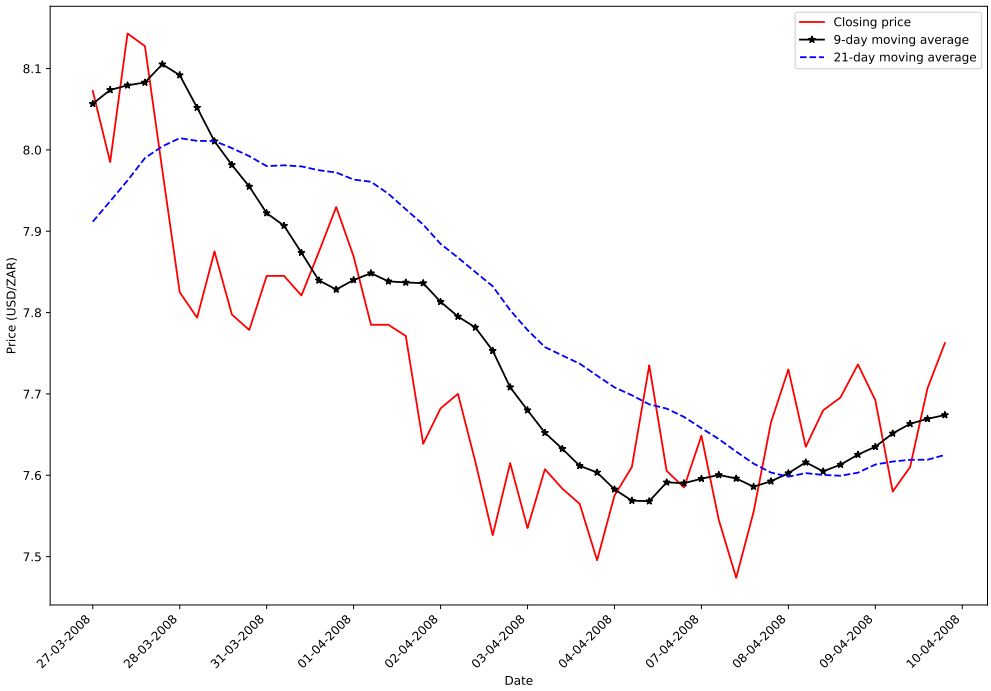


FIGURE 3.1: *An illustration of a 9 and 21-day MA plotted over the closing price of the USD/ZAR exchange rate. A sell signal would be generated at the first crossover point as the 9-day MA crosses below the 21-day MA.*

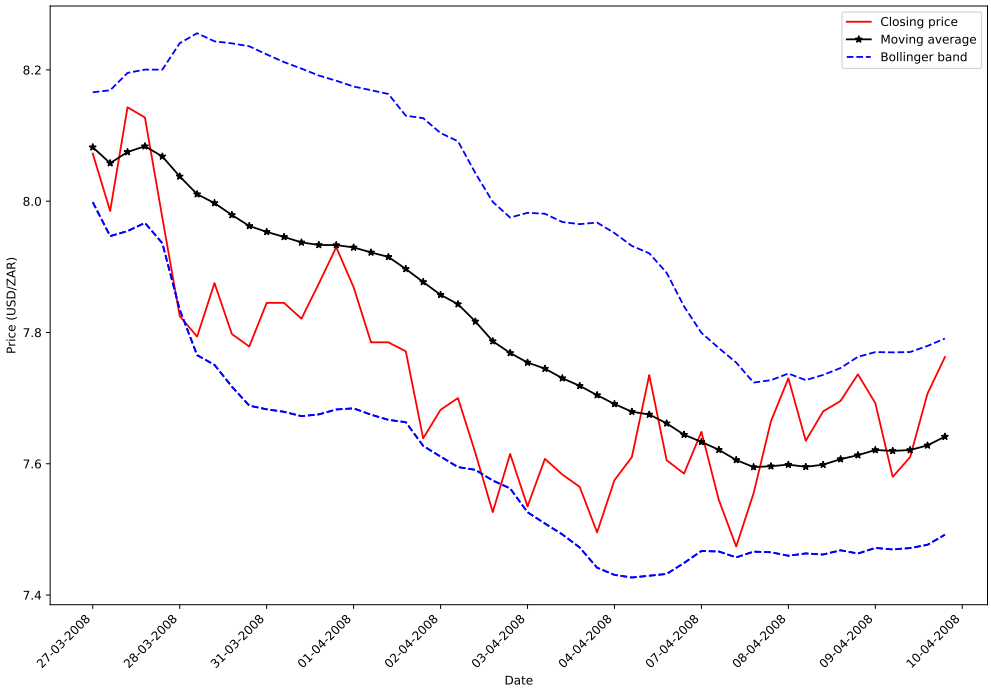


FIGURE 3.2: *An illustration of Bollinger bands plotted over the closing price of the USD/ZAR exchange rate.*

3.1.3 Relative strength index

The Relative Strength Index (RSI), according to Colby [18], is "one of the most popular price momentum indicators". What makes RSI powerful is its ability to quantify momentum by only considering the closing price of the specified financial instrument. In mathematical terms, the RSI is described as

$$\mathrm{RSI}_t = 100 - \frac{100}{1 + \mathrm{RS}_t},$$

where RS is the absolute value of the ratio of n period average gains divided by n period average losses and t is the given time period. The RSI is most commonly used to indicate extended price movements in a specific direction. When the price movement is extended to the upside, it is known as "overbought" and when it is extended to the downside it is known as "oversold". This is done by looking at the RSI values. If the RSI is above a value of 70, it indicates that the market is "overbought" and positive momentum is decreasing — expect a possible decrease in the market. When the RSI is below a value of 30, the market is "oversold" and negative momentum is decreasing — expect a possible increase in the market. However, in Figure 3.3 the RSI never reaches either of these values and drifts between the 40 and 60 levels. This would indicate that the market is moving "sideways" and no significant price movement is occurring.

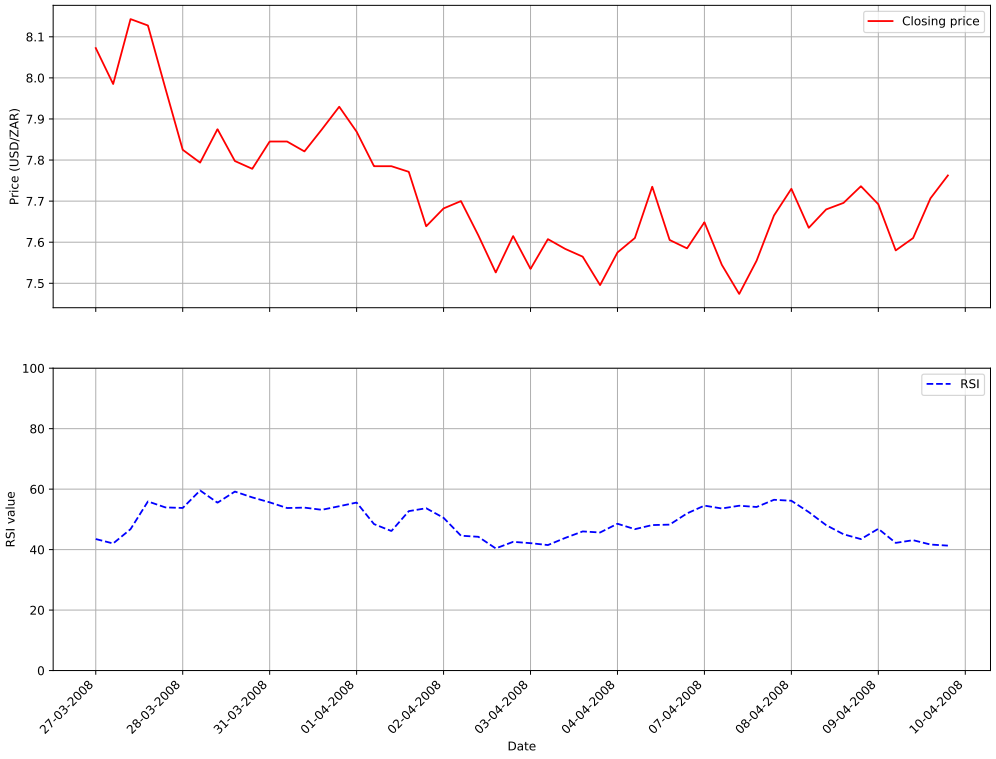


FIGURE 3.3: *An illustration of a 14-day RSI plotted below the closing price of the USD/ZAR exchange rate.*

3.1.4 Heiken-Ashi candlesticks

According to Kuepper [30], Heiken-Ashi (HA) candlesticks make traditional candlestick charts more readable and ease the process of identifying trends. They are very similar to regular candlesticks, but are calculated differently. Mathematically, they are represented as

$$\begin{aligned} \mathrm{HC}_t &= \frac{1}{4}(\mathrm{Open} + \mathrm{High} + \mathrm{Low} + \mathrm{Close}), && (3.1)\\ \mathrm{HO}_t &= \frac{1}{2}(\text{Previous Open} + \text{Previous Close}), && (3.2)\\ \mathrm{HH}_t &= \texttt{max}(\mathrm{Open}, \mathrm{High}, \mathrm{Close}), && (3.3)\\ \mathrm{HL}_t &= \texttt{min}(\mathrm{Open}, \mathrm{Low}, \mathrm{Close}), && (3.4) \end{aligned}$$

where HC_t is the Heiken-Ashi closing price for a time period t, HO_t is the Heiken-Ashi opening price for a time period t, HH_t is the Heiken-Ashi high price for a time period t and HOL_t is the Heiken-Ashi low price for a time period t. In Figure 3.4 the graph on top makes use of normal candlesticks whereas the graph on the bottom uses the HA candlesticks. A major difference between the two graphs is that the HA graph looks smoother and that candles tend to have the same colour when price is trending. This differs when looking at the normal candlesticks where candles change colour despite the overall trend of the price.

FIGURE 3.4: *An illustration of normal candlesticks (top plot) compared to Heiken-Ashi candlesticks (bottom plot).*

3.1.5 Ichimoku kinki cloud

The Ichimoku kinko cloud "is a collection of technical indicators that show support and resistance levels, as well as momentum and trend direction" [33]. The entire indicator consists of 5 lines, of which 4 are moving averages and one is a lagging line, and a cloud. These lines as well as the cloud are defined as follows:

1. The Tenkan-sen (conversion line) is a 9-day moving average.

2. The Kijun-sen (base line) is a 26-day moving average.

3. The Senkou Span A (Leading Span A) is plotted 26 days ahead and can be defined mathematically as
$$\text{Senkou Span A} = \frac{\text{Kenkan-sen} + \text{Kijun-sen}}{2}.$$

4. The Senkou Span B (Leading Span B) is the average between the highest and lowest price over the past 52 days. It is also plotted 26 days ahead and can be defined mathematically as
$$\text{Senkou Span B} = \frac{52\text{High} + 52\text{Low}}{2}.$$

5. The Chikou Span (Lagging Span) is the closing price plotted 26 days behind the previous closing price.

6. The cloud represents the difference between the Leading Span A and B and is a shaded region on the graph (light blue shaded area in Figure 3.5). It is plotted 26 days ahead and indicates possible support and resistance points.

There are several strategies that could be used when looking at Ichomoku clouds. A simple strategy to identify a trend is by looking at the position of the closing price in relation to the cloud. If the closing price is below the cloud, a downward trend can be identified and when the closing price is above the cloud, an upward trend can be identified. When Figure 3.5 is considered, most of the trend is considered to be downwards as the closing price (solid red line) is below the Cloud (shaded light blue region).

3.1.6 Implied volatility and risk reversal

Implied volatility and risk reversal are used to analyse market information and sentiment. Implied volatility, according to Ganti [20], "is a metric that captures the market's view of the likelihood of changes in a given security's price". This means that implied volatility provides a glimpse into the general gut feeling of the market when the specific security is considered. In a bearish market, the implied volatility will increase as the price movement of a security is believed to decline with time (higher risk for investors). However, the opposite is true for a bullish market — implied volatility will decrease as the security's price is believed to increase (less risk for investors). Consider Figure 3.6 where a stock is trading at ZAR50 and the implied volatility is 20%. This would imply that the overall gut feeling of the market would conform to a one standard deviation move of $\pm$ZAR10 (since 20% of ZAR50 is ZAR10) over the next 12 months of the stock - an expected stock price of either ZAR40 or ZAR60.

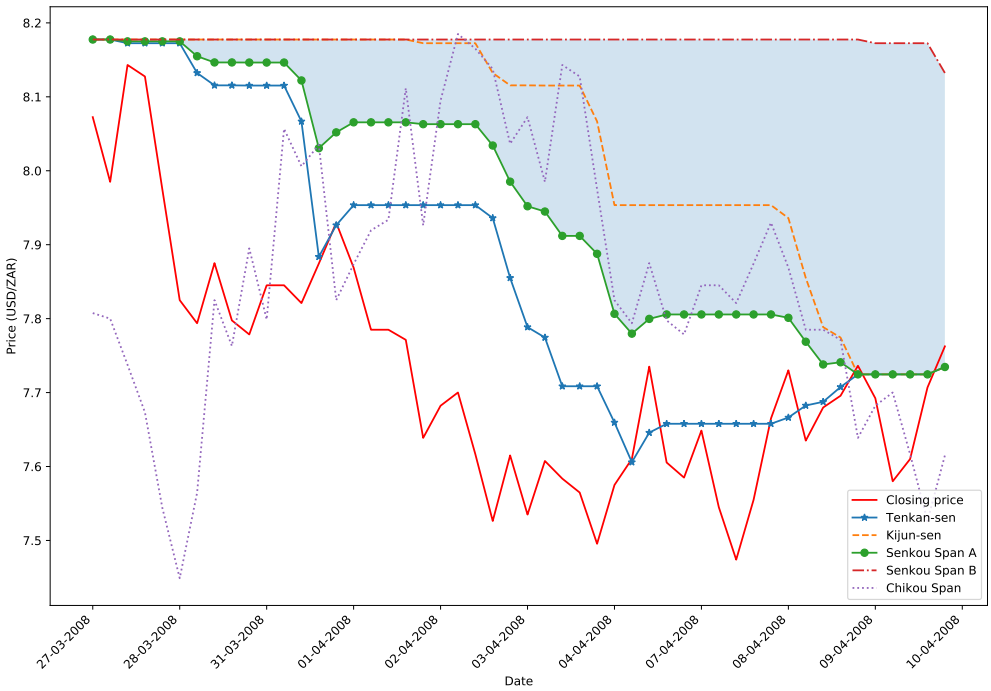


FIGURE 3.5: *An illustration of Ichimoku Kinko Clouds plotted over the closing price of the USD/ZAR exchange rate.*

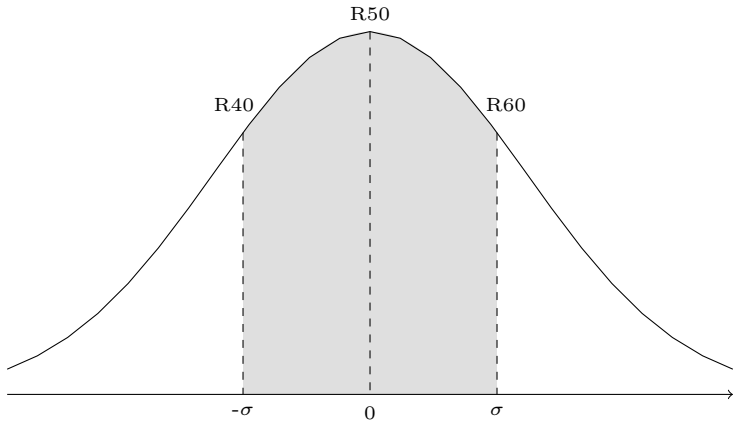


FIGURE 3.6: *An illustration of how implied volatility effects the exchange rate or the price of a stock.*

Risk reversal is the difference in implied volatility of out of money (OTM) calls and OTM puts[1] (OTM describes option contracts that have no intrinsic value). It indicates whether investors believe the market will increase or decrease [44]. If the risk reversal is positive it means that more investors are betting on the rise of the market than against. The opposite is true for a negative risk reversal.

3.2 Convolutional neural network

The visual cortex is a region in the brain that receives and processes visual information obtained from the eyes (specifically the retinas). Hubel and Wiesel [26] did an experiment with cats where they found that several neurons within the visual cortex react only to visual stimuli, forming a local receptive field. Their study also showed that some of the neurons within the receptive field react only to images of horizontal lines, while others react only to lines with different orientations. These neurons can be thought of as low-level neurons which, when combined, form higher-level neurons capable of identifying complex patterns [21]. This inspired neocognitron, a hierarchical, multilayered artificial network, which evolved over time into Convolutional neural networks (CNNs).

CNNs are specialised neural networks that require grid-like data [24]. Image data, which is just a 2-dimensional grid of pixels, is the most common data used. However, restructuring time series data into a 1-dimensional grid can also be used. It follows that with a different structure of data required, the architecture of a CNN differs from a regular neural network. CNNs typically have an input layer, convolutional layer, rectified linear unit (ReLu) layer, pooling layer and fully connected layer [28].

Figure 3.7 shows the typical structure of a CNN. The input layer simply receives the input data. The convolutional layer is the building block of a CNN and is based on a mathematical operation known as a convolution. Goodfellow *et al.* [24] describe a convolution as "an operation on two functions of a real-valued argument" that aims to lessen noisy data by using a weighted average operation. Each convolutional layer is made up of filters (a set of weights) which are combined to form feature maps. Each feature map can detect different patterns and, when combined, form a convolutional layer. During training, the CNN will search for filters that are the most useful, and combine them to form patterns with increased complexity [21].

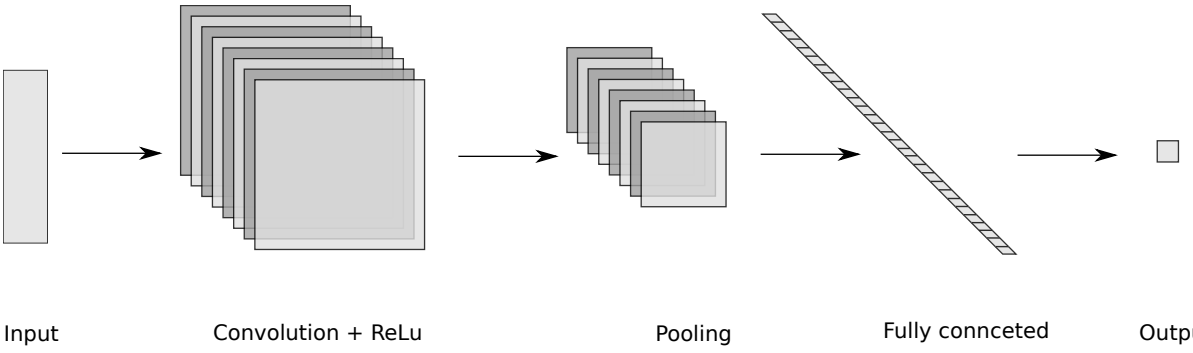


FIGURE 3.7: *A schematic representation of the structure of a typical Convolutional Neural Network.*

The ReLu layer, according to Joshi [28], is required to "add non-linearity to the network" and assist in generalisation. It helps decide whether a specific feature is present at a given location in

[1] A call option gives the holder the right to purchase the selected currency at a given price and a put option gives the holder the right to sell the selected currency at a given price [2].

the data. The pooling layer typically identifies the largest values in the feature maps, resulting in dimensionality reduction of the output from the previous layer [28]. It therefore aids in the detection of the most prominent features which will then be given to the next layer. The final fully connected layer then computes the output of the NN.

As CNNs are specialised in grid-like data, a special type of NN is required when sequential data is considered. These NNs are known as Recurrent Neural Networks.

3.3 Long-short term memory

A Long-short term memory (LSTM) network is a special type of Recurrent Neural Network (RNN). Therefore an understanding of the functionality of a RNN is required to understand how a LSTM functions. Similar to CNNs which specialise in processing grid-like data, RNNs are specialised in the processing of sequential data [24].

Sequential data refers to data that is structured in a specific order, such as time series data. RNNs are specialised in sequential data as it has a form of memory [21]. This memory, referred to as sequential memory by Nguten [34], can be illustrated using the months of the year. Saying the months from start to finish is an easy task. However, saying the months backwards is surprisingly difficult. This is due to the brain remembering the sequence of the months from start to finish. Starting from a random month, "August" for illustration purposes, and then finishing the months of the year from that selected starting month is also challenging initially. However, it becomes easier once the brain has adapted and remembers the rest of the sequence. Sequential memory of a RNN functions in a very similar way.

A RNN is made up of multiple RNN cells, which has 4 main components: the input x_t, the activation function σ, the short-term state (referred to as either "sequential memory" or "hidden state") h_t and the output $\hat{y}_t$. Figure 3.8 illustrates this basic structure of a single RNN cell and shows how the hidden state loops around and is used again. Taking one step deeper into the structure of the RNN cell and considering a sequence as input, the way in which a RNN cell passes the short-term state to the next cell is shown in Figure 3.9. This is referred to as unrolling the network through time [21].

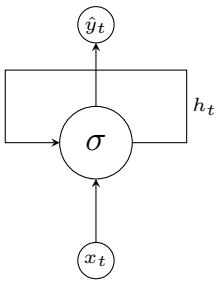


FIGURE 3.8: *An illustration of a single RNN cell.*

However, a RNN has a major flaw in its design, namely the vanishing gradient problem. When a RNN is trained during back-propagation, the gradient (used to update weights within the network) for successive layers become exponentially smaller when a previous layer receives a small weight update. This results in minor weight updates, causing early layers in the network not to learn. The problem of a vanishing gradient inevitably results in a RNN not learning the long-term dependencies between inputs, thus having a short-term memory.

A LSTM network, according to Géron [21], "will detect long-term dependencies in the data"

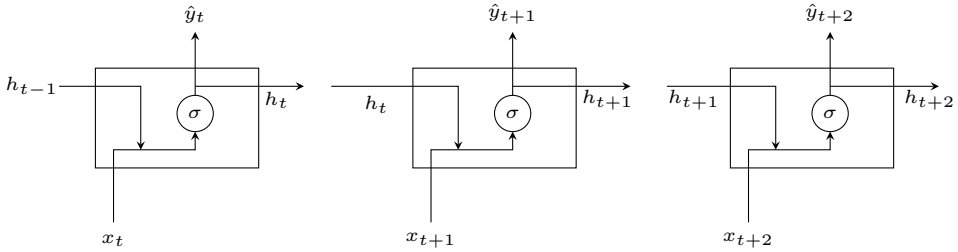


FIGURE 3.9: *An illustration of how the hidden state is passed to the next RNN cell when a sequence is given as input.*

and deliver on the promise made by a RNN of learning the temporal and contextual information within data [13]. LSTMs solve the problem of short-term memory, and do so by using gates. A representation of the internal mechanics of a LSTM cell is shown in Figure 3.10, and shows the three different gates, within the dashed rectangular boxes, found within a LSTM cell.

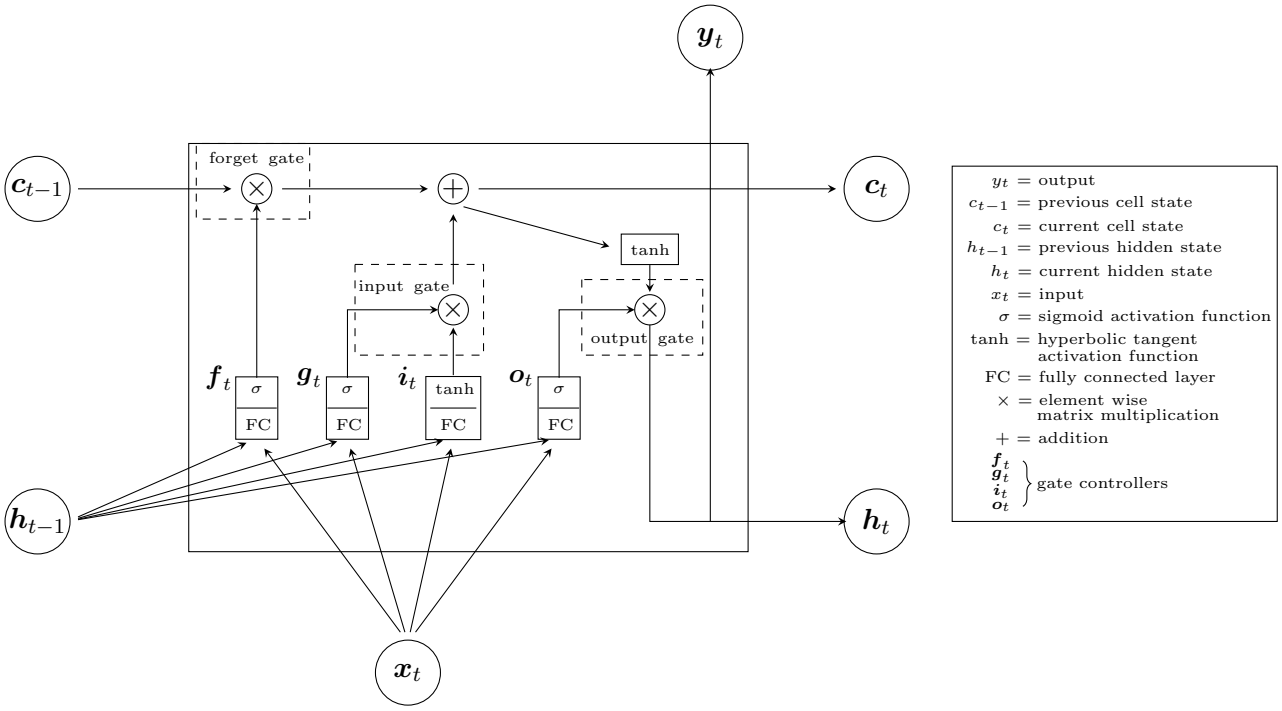


FIGURE 3.10: *A schematic representation of a LSTM cell as illustrated by Géron [21].*

The gates govern information flow within the cell by altering the cell state (long-term memory). Gates determine which data is relevant and has meaning, and which is irrelevant and can be thrown away [13]. The cell state is the main reason why later time steps can access information from previous time steps, and can be thought of as the "highway" of information [34]. For this reason, all gates have access to the cell state and could potentially alter it. The forget gate decides which information should be thrown away (forgotten) from the cell state [21]. This is achieved by looking at the output values. Values are between 0 and 1, with values closer to 0 meaning "forget" and values closer to 1 meaning "remember".

Similar to an individual paying expenses every month, if income ceases, nothing will be left to pay future expenses. The input gate can therefore be thought of as an individual receiving a loan, injecting capital and allowing the flow of money to continue. The input gate therefore

controls which information is added to the cell state, replacing old information. Similarly to the forget gate, this is done by looking at the output values. A value close to 0 will not be added whilst a value closer to 1 will be added.

Once an individual has received money, their bank account value increases and they can pay expenses again. However, to prevent impulsive spending, the individual could remember what they needed to purchase that month (groceries, gifts etc) and make a note of it. The final output gate, works in a similar way. It produces two outputs, an updated hidden state and a prediction. The new hidden state can be thought of as the individual taking note of all the expenses that needed to be paid, and the prediction can be thought of as the money left over at month-end.

3.4 Data processing

The baseline dataset (to which additional features were added) used for both training and testing consisted of daily historical price data of the USD/ZAR currency pair. Training and validation data ranged from 1 January 2014–31 December 2018, of which 90% was used for training and 10% was used for validation. The data used for the test set ranged from 1 January 2019–31 May 2019. Preprocessing of the data entailed removing any and all samples which had the open, high, low or closing price missing (of which there were none), calculating all relevant input features for the conducted test, and normalising and transforming the data into a range between 0 and 1 using the scikit-learn MinMaxScaler method [6]. Normalisation was done for data used by both the statistical forecasting methods and neural networks.

The transformed data would then be sliced into two vectors. The first would represent the $n \times 1$ input vector $\boldsymbol{x}$, where n is calculated as the number of features multiplied by the number of previous days as input. Each test would have the number of previous days range from 1–5. Testing a different number of previous days as input allows deeper insight into how much information the models require when making predictions. The second vector would represent the 1×1 desired output vector (the label). This allows the time-series forecasting problem to be considered as a supervised learning problem.

However, to restructure the problem as a supervised learning problem, the labelled data has to be the closing price at time $t + 1$. Consider Table 3.1(a) where the price of an item is given for 5 days. Restructuring this data into useable labelled data for a supervised learning problem would then require that the price of the item at day t, be shifted to the previous day $t - 1$. This is seen in Table 3.1(b) where a label has been added to the data.

Day	Price (ZAR)
1	10
2	20
3	30
4	40
5	50

(A) *An example of time series data.*

Day	Price (ZAR)	Label
1	10	20
2	20	30
3	30	40
4	40	50
5	50	/

(B) *An example of labelled time series data.*

TABLE 3.1: *An example of restructuring time series data into a labelled dataset suitable for supervised learning.*

Walk-forward validation was used to simulate real-life predications and use thereof. When predictions were made with the test set, only the previous n days of data would be available

(simulating a real-life scenario). Once a prediction was made for the next day, the real data for the predicted day would then be appended to the test set to use for the next prediction. This process ensures that data, that would otherwise not be available in real-life, does not leak into the test set and compromise the validity of predictions.

3.5 Modelling

Three NN models were implemented and tested. All neural networks were based on a LSTM network, but with the second and third models having an additional CNN added to the existing LSTM architecture. The second and third models are therefore Encoder-Decoder networks, due to the placement of the CNN, in which the CNN acts as the encoder and the LSTM the decoder. This is to investigate if the CNN can extract and amplify any salient features which the LSTM can then use to make better predictions.

All models were trained and tested on a computer running Ubuntu 18.04.4 LTS 64-bit operating system equipped with a 3.7GHz AMD Ryzen 7 2700X processor, ZOTAC GeForce GTX 1080 Ti AMP Extreme graphics card and 16GB 3200MHz DDR4 RAM. The models were implemented in Python 3.6 [5] using the NVIDIA cuDNN [4] and Keras [3] libraries.

The tests conducted as well as their respective input features (chosen with the help of a professional foreign exchange trader) can be seen in Table 3.2 — a complete table indicating all input features used for each test can be found in Table A.73. Test 1 was used as the baseline to which all other tests were compared and used the raw open, high, low and closing price of the USD/ZAR exchange rate as features. All tests succeeding Test 1 had their features added to the existing features of Test 1 and then removed for the next test. As the number of input time steps (previous days) also affects predictive performance, the number of previous days given as input would vary from 1 previous day to 5 previous days.

Test	# Features	Features
1	4	open, high, low, close (OHLC)
2	6	(OHLC) + 9MA, 21MA
3	7	(OHLC) + price change
4	12	(OHLC) + RSI
5	17	(OHLC) + Heiken Ashi
6	24	(OHLC) + Ichimoku Kinko Cloud
7	8	(OHLC) + Bollinger Bands
8	11	(OHLC) + 3 month implied volatility
9	47	(OHLC) + implied volatility, risk reversal
10	12	(OHLC) + 1^{st} and 2^{nd} difference
11	102	All features
12	31	Features determined by PCA

TABLE 3.2: *A table showing the input features for all the different tests that were conducted.*

For the scope of this study, the main focus of tuning the neural networks was placed on architecture. Almost all other hyper parameters that the neural networks require were kept at default values with the exception of several chosen hyper parameters being changed but kept constant throughout testing. This decision was made due to time constraints and preventing the scope of the study to widen substantially. When referring to parameters that were "changed but kept constant", it refers to those parameters maintaining the same chosen value throughout

all tests that were conducted. This allows consistency to be achieved throughout the tests. If hyper parameters were changed inconsistently throughout tests, a true comparison of the results would be invalid as one test could have received a better hyper parameter value than another. The hyper parameters that were chosen and kept constant are as follows:

1. The Rectified Linear Unit (ReLu) activation function for all layers except the final output layer which had a linear activation function. The ReLu activation function was chosen due to its state-of-the-art performance in literature.
2. The number of epochs to train the models was set at 30, allowing manageable run times for training.
3. A dropout layer was used between the CNN and LSTM neural network in the CNN-LSTM model, aiding the prevention of overfitting.
4. Early stopping was used in the training of all models to prevent overfitting.
5. A batch size of 32 (the general default value) and learning rate of 0.001 (the median of the default range) was selected.
6. All models used the Adam optimiser as it allows fast convergence and was found to be the best overall optimizer to use [41].

Accuracy was used as the metric to evaluate and compare the predictive power of the different networks. A prediction was considered accurate if the prediction fell within a threshold of R0.02 (chosen with the help of industry professionals) from the actual closing price. However, if networks had similar accuracies then the mean squared error (MSE) was used for evaluation.

The LSTM network was chosen as a baseline as it performs, on average, better than a RNN or ANN for time series predictions [8, 25, 40, 48, 49]. Adding a CNN to an LSTM network to form a hybrid network was inspired by the work done by Ni *et al.* [35] and Hiransha *et al.* [25]. They found that a CNN outperformed a RNN, LSTM and MLP and suggested that future work could be done on a hybrid network. It was therefore decided that two hybrid models would be tested in conjunction with the baseline LSTM model: a CNN-LSTM and multi-head CNN-LSTM model.

3.5.1 LSTM network

The LSTM network consists of an input layer, l hidden layers and an output layer (as seen in Figure 3.11). The dotted rectangle in Figure 3.11 represents where the architecture would be modified if the number of hidden layers or LSTM cells within each layer were to be modified.

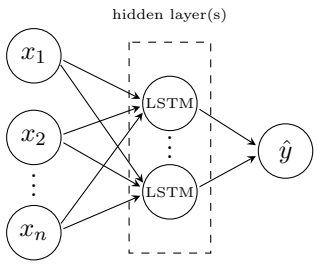


FIGURE 3.11: *A schematic representation of the implemented LSTM model.*

3.5.2 CNN-LSTM network

Since Hiransha *et al.* [25] do not specify how the suggested hybrid model would be built, the first approach was to add a CNN prior to the LSTM network. The CNN-LSTM would consist of an input layer, a CNN followed by a LSTM neural network and finally an output layer. All the input features would be given to the CNN which would then potentially extract any salient features from the inputs. The CNN would then produce an output which the LSTM would use an input and make a prediction. A basic representation of the architecture can be seen in Figure 3.12 with a more detailed representation shown in Figure 3.13.

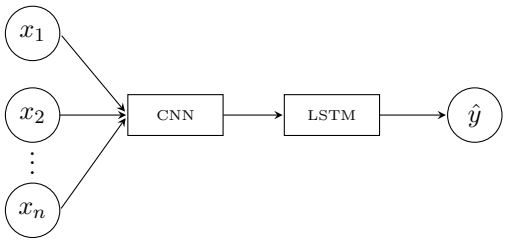


FIGURE 3.12: *A simplified schematic representation of the implemented CNN LSTM model.*

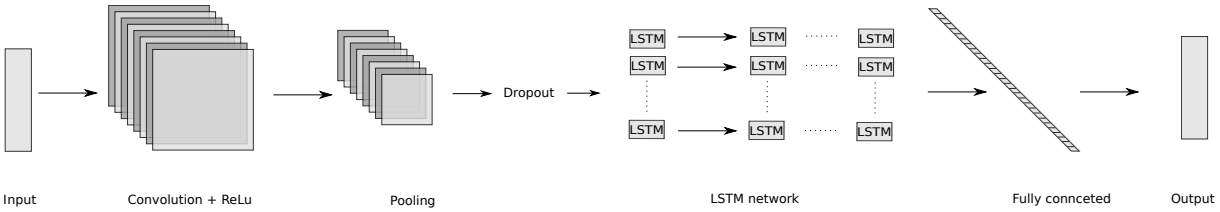


FIGURE 3.13: *A detailed schematic representation of the implemented CNN LSTM model.*

3.5.3 Multi-head CNN-LSTM network

The second approach was to build a hybrid model by using a CNN for each input, creating the multi-head aspect of the model. The output from the multi-head CNN is then combined to form the input for the LSTM model. It consists of an input layer, a multi-head CNN layer for each input, a LSTM layer and finally an output layer. A basic representation of the architecture can be seen in Figure 3.14 with a more detailed representation shown in Figure 3.15.

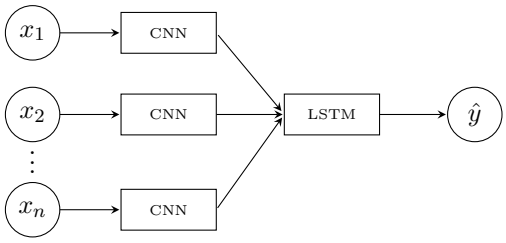


FIGURE 3.14: *A simplified schematic representation of the implemented multi-head model.*

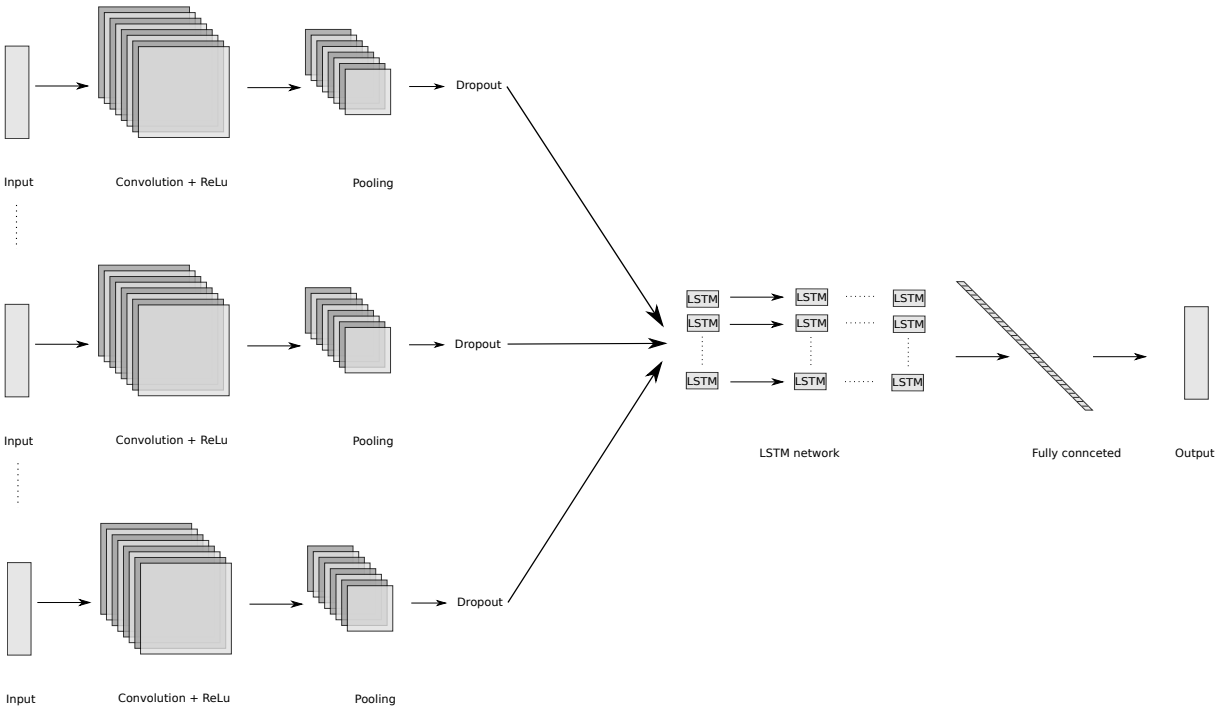


FIGURE 3.15: *A detailed schematic representation of the implemented multi-head model.*

CHAPTER 4

Principal component analysis

Contents

4.1 PCA criterion 31
4.2 Applying PCA 32

If the available input data for a problem has a large number of features, it could lead to an unacceptable increase in computational time required as the complexity of the problem increases. A method to reduce the number of features for the problem is by using principal component analysis (PCA). PCA, according to O'Rourke and Hatcher [36] "is a variable reduction procedure" and can be used to remove any redundant variables that are correlated to one another. The final outcome of PCA is a number of principal components which best explain the variance within the data.

A principal component (PC) is linear combination of all the variables in the data. Each variable is weighted and indicates the contribution it plays when calculating the PC. Mathematically, O'Rourke and Hatcher [36] describes the calculation of a PC as

$$PC_i = b_{i1}x_1 + b_{i2}x_2 + \ldots + b_{in}x_n, \tag{4.1}$$

where PC_i is the i^{th} principal component, b_{in} is the weight for feature n in PC_i and x_n is feature n.

4.1 PCA criterion

In this study, PCA was first applied by using SAS [1] to both analyse and decide which features should be retained. All linearly dependent variables were removed after which the VARIMAX rotation was used to produce orthogonal PCs. The following criterion, from O'Rourke and Hatcher [36] as well as Schönrock-Adema *et al.* [42], were used to determine the number of retained components:

1. The eigenvalue-one criterion — components with an eigenvalue greater than or equal to one will be retained.

2. The scree test — a plot of all the eigenvalues associated with each component. If a sudden drop off in the magnitude of the eigenvalues occur, it would indicate that all components prior to the drop off would be retained.

3. Kaiser's Measure of Sampling Adequacy (KMA) — only features with a KMA value greater than equal to 0.75 should be retained.

4. Only features with a factor loading greater than or equal to $|0.3|$ should be retained.

5. Components that account for more than 5% of the variance will be retained.

4.2 Applying PCA

Before PCA was applied, all linearly dependent variables were removed. This reduced the total number of features, initially 102, to 43. An additional 9 features were removed due to KMA values below 0.75, reducing the features down to 34. Lastly, 3 features had a factor loading smaller than $|0.3|$, reducing the number of features to a final total of 31.

The scree plot in Figure 4.1 as well the eigenvalue-one criterion applied to Table 4.1 indicate that 6 components should be extracted. However, an interpretability criterion from Schönrock-Adema *et al.* [42] state that "a given component contains at least three variables with significant loadings". Component 6 does not contain at least 3 features with significant loadings and will therefore not be considered. The remaining 5 extracted components have a cumulative percent of variance of 71.48% and can be grouped as (1) average price, (2) risk reversal, (3) implied volatility, (4) Ichimoku Kinko Cloud, and (5) the change in closing price.

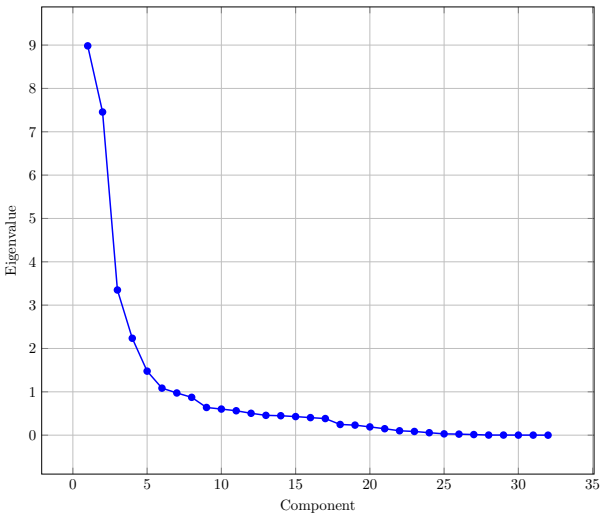


FIGURE 4.1: *A scree plot of the eigenvalues obtained through PCA.*

The interpretability of the principal components and which features are present in each component gives insight into the data and what relationships could exist between the variables. According to Schönrock-Adema *et al.* [42], in order to interpret the components there are 2 main criteria in addition to the criterion mentioned in Section 4.1. The first is that all the features within the component share the same conceptual meaning, and the second is that each feature within the component should represent different constructs. All 5 components satisfy both criteria and could be described as follows:

Component	Eigenvalue
1	8.97474951
2	7.39091250
3	3.34014657
4	2.10927077
5	1.28009193
6	1.08026061
7	0.97238709
8	0.86061492
9	0.60952051
10	0.56836799

TABLE 4.1: *A table showing the input features for all the different tests that were conducted.*

1. The average price: examples of features in this component include a 9 and 21-day moving average, and Bollinger bands.

2. Risk reversal: the change in 3, 6 and 9 month risk reversal values are examples of features.

3. Implied volatility: the change in 3, 6 and 9 month implied volatility values are examples of features.

4. Ichimoku Kinko Cloud: the difference between the kijun sen and senkou span A is an example of a feature.

5. The change in closing price of the exchange rate: the change in closing price is an example of a feature.

CHAPTER 5

Statistical forecasting models

Contents

5.1 Naïve 35
5.2 Simple exponential smoothing 36
5.3 Multiple regression 36
5.4 Naïve results 37
5.5 Simple exponential smoothing results 37
5.6 Multiple regression results 39

Statistical forecasting methods, when compared to machine learning techniques, could be considered as the most basic form of forecasting. For this reason, statistical methods provide an easy way to compare the performance of several forecasting models. The remainder of this chapter will discuss the relevant statistical models that were implemented as well as their results.

5.1 Naïve

A naïve forecast is the most rudimental form of forecasting, where an occurrence in the current period will repeat itself in the next [46]. Think of it in terms of weather, if today is a warm summers day, a logical conclusion would be that tomorrow will also be a warm summers day. A naïve forecast has the exact same logical conclusion and can be described mathematically as

$$\hat{y}_t = x_{t-1}$$

where $\hat{y}_t$ is the forecasted value for time t and x_{t-1} represents the observed value at time $t-1$. As predictions rely solely on the previous observation, a naïve forecast can be done on any type of time series data.

5.2 Simple exponential smoothing

Simple exponential smoothing (SES) is similar to a moving average (forecasted value is an average of past values) with the distinguishing factor being that forecasted values are now a "weighted average of all available previous values" [46]. A SES model can be expressed mathematically as

$$\hat{y}_{t+1} = \alpha x_t + (1 - \alpha)\hat{y}_t, \tag{5.1}$$

where

$$\hat{y}_{t+1} = \text{forecasted value for time } t + 1, \tag{5.2}$$
$$\alpha = \text{smoothing constant, } \alpha \in (0, 1), \tag{5.3}$$
$$x_t = \text{actual value at time } t, \tag{5.4}$$
$$\hat{y}_t = \text{forecasted value at time } t \tag{5.5}$$

A limitation to SES is that it should only be applied to stationary data (no trend or seasonality present). However, with the introduction of an error term, SES gains a "learning" element by rewriting equation (5.1) as

$$\hat{y}_{t+1} = \hat{y}_t + \alpha(x_t - \hat{y}_t), \tag{5.6}$$

where $x_t - \hat{y}_t$ is the error between the actual and predicted value and the "learning" element. Equation (5.6), with the addition of the error term, allows small corrections to be made to predictions and therefore suggests that SES "learns" as predictions are made [46]. Learning takes place in the form of either increasing the predicted value $\hat{y}_{t+1}$ when $x_t > \hat{y}_t$ or decreasing $\hat{y}_{t+1}$ when $x_t < \hat{y}_t$.

5.3 Multiple regression

Naïve and SES models only consider the independent variable when making forecasts. A drawback to this method is that subtle relationships could exist between independent variables and overlooking these relationships could lead to a decline in forecasting performance. Multiple regression (MR) takes these possible relationships into account by forecasting a dependent variable, $\hat{y}$, as a function of several independent variables $(x_1, \ldots, x_n)$ [46]. MR can be expressed mathematically as

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n, \tag{5.7}$$

where $\hat{y}$ is the predicted value, b_0 is the intercept and $b_1, \ldots, b_n$ are the regression coefficients which represent the change in $\hat{y}$ when the relevant independent variable is changed by one unit. Due to the lack of multicollinearity between features identified by PCA in Chapter 4, these features were used as independent variables to build the MR model.

All the statistical forecasting models were trained on the entire training and validation dataset mentioned in Section 3.5. Training, when considering the statistical forecasting models, refers to both the fitting and parameter optimisation (α in the case of SES) for that model. Testing of the forecasting models was done on the test set, also mentioned in Section 3.5. All statistical forecasting models were implemented, trained and tested in Python.

5.4 Naïve results

The predictions made by the Naïve model, shown in Figure 5.1, achieved an accuracy of 27.2% and a MSE of 0.011393858. As expected, the predictions follow the trend of the actual values. However, due to the one day lag in the predictions, it resulted in poor predictions on several occasions.

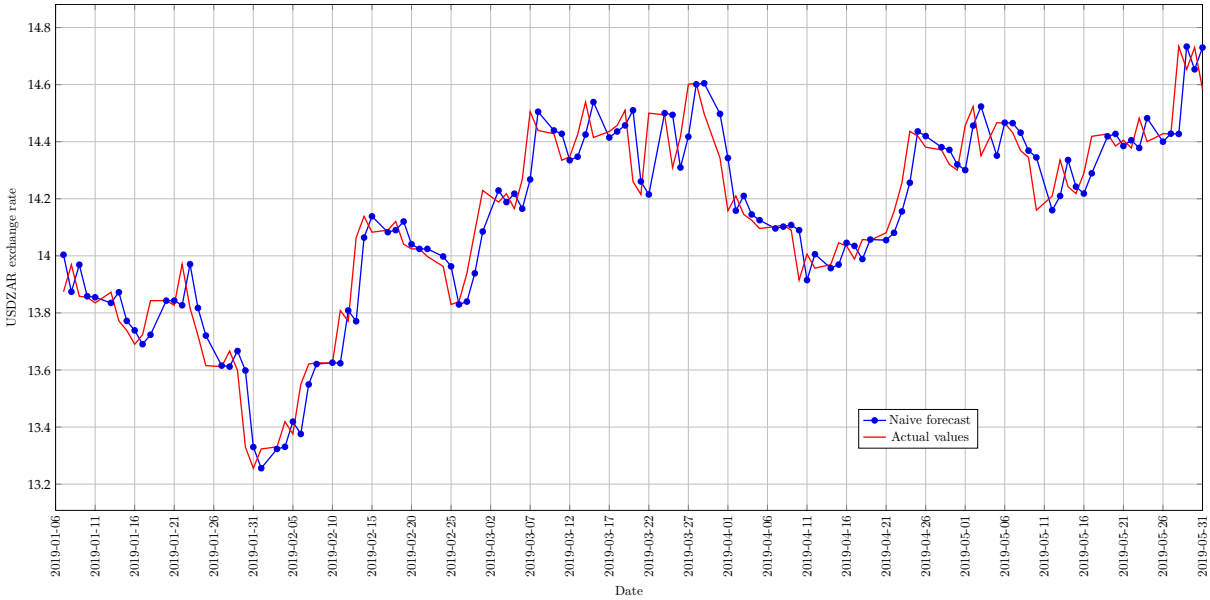


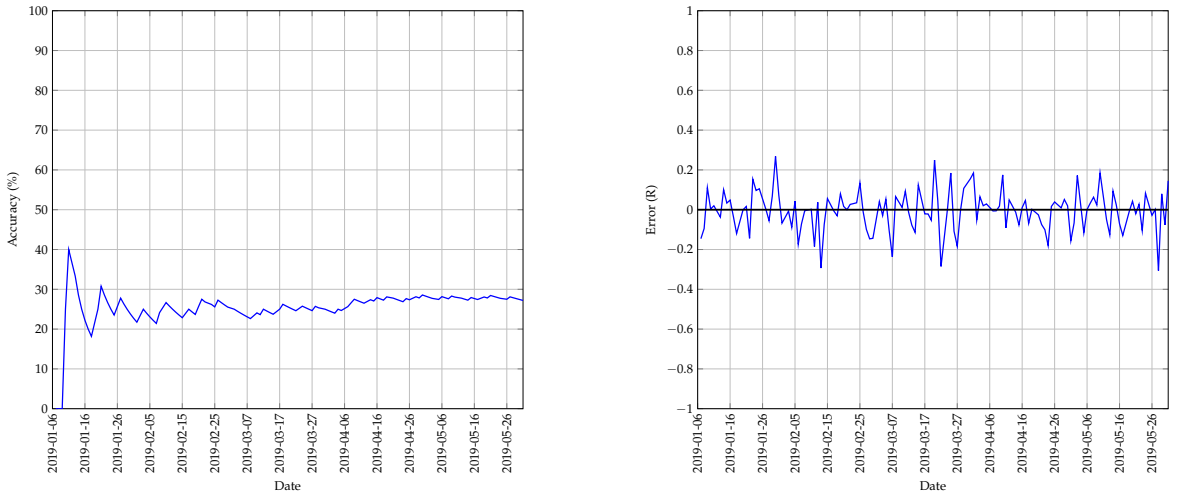
FIGURE 5.1: *The predictions made by the Naïve forecast (blue line) compared to the actual values (red line).*

The change in accuracy, shown in Figure 5.2(a), provides a deeper insight into the relationship between predictions and actual values. After the initial predictions, the accuracy jumps to 25% and then to 40% after which it decreases and fluctuates in the range of 20%–30%. However, the change in error as predictions are made, shown in Figure 5.2(b) , can provide clues as to why the accuracy does no increase steadily. The change in error fluctuates between $|0.2|$ with several outliers breaking past this level. As the Naïve forecast is just predicting the previous closing price, the change in error shows the volatility of daily price movement. This volatile and unpredictable nature of price movement in foreign exchange is the reason that the forecasting error remains fairly constant over time.

5.5 Simple exponential smoothing results

After fitting the SES model, $\alpha = 0.02$ was obtained. The resulting SES model,

$$\hat{y}_{t+1} = \hat{y} + 0.02(x_t - \hat{y}_t), \tag{5.8}$$



(a) *The change in accuracy of the Naïve model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the Naïve model whilst using a walk-forward validation approach on the test set.*

FIGURE 5.2: *A graphical representation of how the accuracy and error of the Naïve model changed whilst making predictions on the test set.*

was used to make the predictions obtained in Figure 5.3 and obtained an accuracy of 5.6% and MSE of 0.117983748. Although the prediction accuracy was very poor, the SES model does seem to model the underlying trend. This is expected as moving averages remove volatility and smooth price fluctuations. However, the Naïve forecast outperforms the SES model with an increase of 21.6% in prediction accuracy and lowering the MSE by 0.10658989.

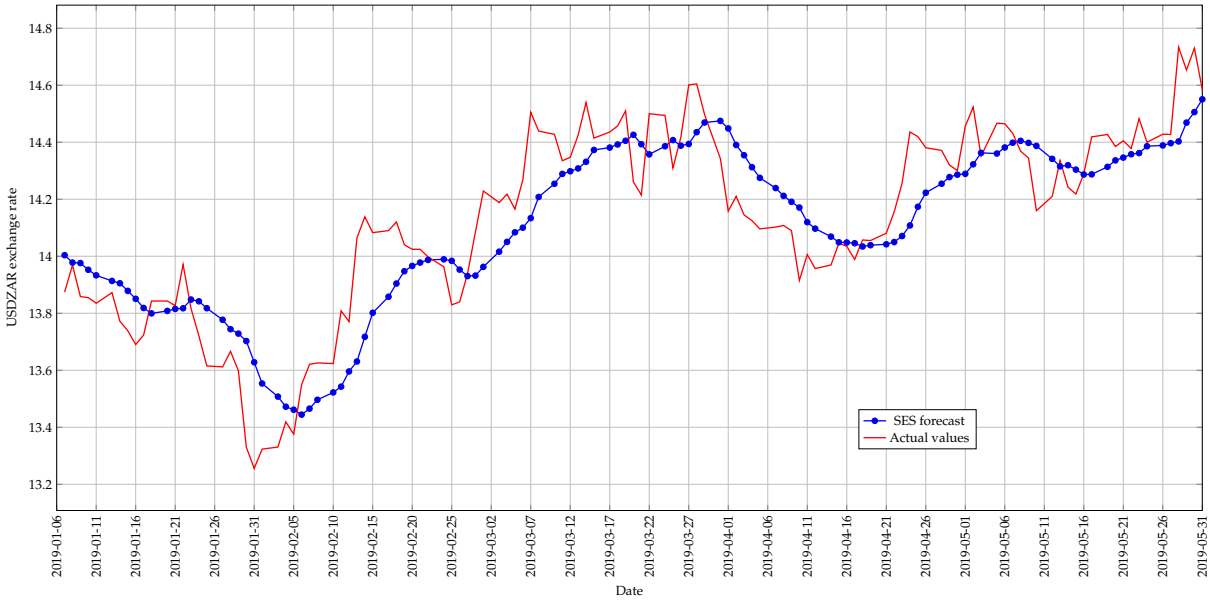
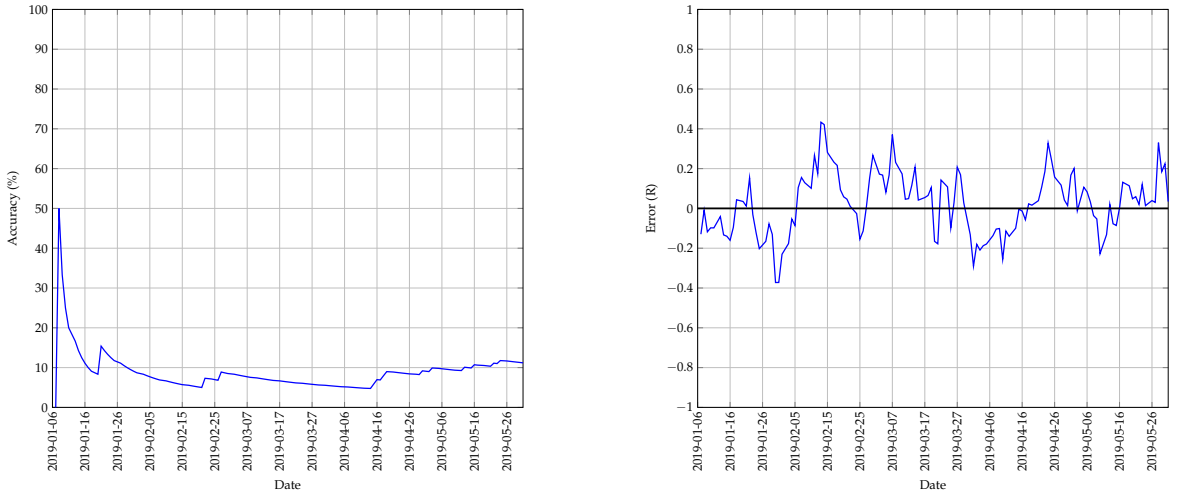


FIGURE 5.3: *The predictions made by the SES model (blue line) compared to the actual values (red line).*

The change in accuracy as predictions are made, shown in Figure 5.4a, illustrates the poor performance of the SES model. The accuracy spikes to 50% initially but then decreases exponentially with further predictions. This is expected when the change in error, shown in Figure 5.4b, is

considered as the error fluctuates between $|0.4|$ — double that of the Naïve forecast. As the SES model is smoothing price movement and not following the actual movement of the price as closely as the Naïve forecast, this increase in error range is expected.



(a) *The change in accuracy of the SES model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the SES model whilst using a walkforward validation approach on the test set.*

FIGURE 5.4: *A graphical representation of how the accuracy and error of the SES model changed whilst making predictions on the test set.*

5.6 Multiple regression results

The MR equation used to make predictions,

$$\begin{aligned} y = {} & 0.0549 + 0.2394x_1 + 0.5856x_2 + 0.141x_[3] - 0.4829x_4 - 0.009341x_5 - 0.2956x_6 \\ & + 0.5922x_7 + 0.007961x_8 + 0.02197x_9 + 0.04533x_{10} - 0.001098x_{11} - 0.02481x_{12} \\ & - 0.01095x_{13} + 0.01252x_{14} + 0.2448x_{15} + 0.2413x_{16} - 0.00907x_{17} + 0.003232x_{18} \\ & - 0.01863x_{19} + 0.00877x_{20} + 0.05892x_{21} - 0.04061x_{22} - 0.01123x_{23} - 0.002185x_{24} \\ & - 0.03643x_{25} + 0.01195x_{26} - 0.003125x_{27} - 0.00005674x_{28} - 0.01782x_{29} \\ & - 0.00176x_{30}, \end{aligned} \tag{5.9}$$

is shown in Figure 5.5 and obtained an accuracy of 30.4% and MSE of 0.0041757. The coefficients for $x_1, \ldots, x_{30}$ are shown in equation (5.9) and displayed graphically in Figure 5.6. A description of the variables is given in Table 5.1. Variables with larger coefficients will have a greater impact on predictions. This deduction can be made as all input data undergoes a transformation as described in Section 3.5. From Figure 5.6, $x_2, x_4, x6, x_7, x_{15}$ and x_{16} had the largest coefficients, and therefore the greatest impact on predictions. These variables represent the high, close, change in closing price, binary increase, RSI, and binary overbought respectively. However, x_7 had the greatest coefficient of 0.5922 which indicates that for every unit the binary increase changes, the prediction will change by 0.5922. This result is somewhat expected as the binary increase variable represents whether the change in closing price is positive or negative. A 1 would indicate that the change in closing price is positive, and a 0 otherwise. Knowing whether the

change in previous closing price was positive or negative could indicate the direction of the next price movement. For this reason, it is somewhat expected that the binary change variable has the greatest influence on prediction. The factor that makes this unexpected is the expectation that a variable, like the RSI which shows price momentum and considers more information than a simple binary variable, would have the highest influence on predictions.

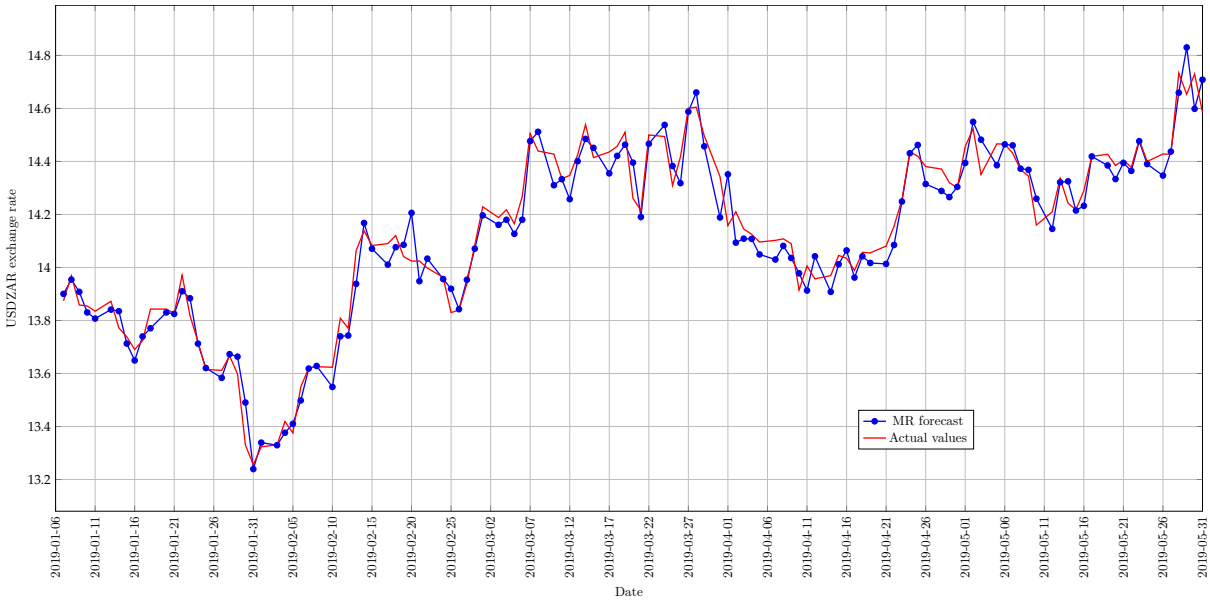


FIGURE 5.5: *The predictions made by using MR (blue line) compared to the actual values (red line).*

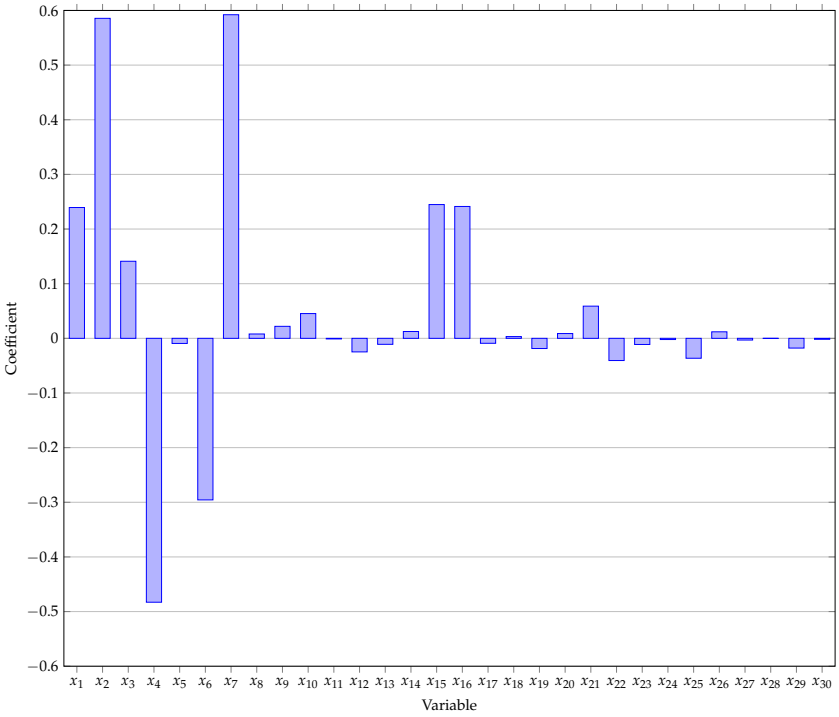


FIGURE 5.6: *An illustration of which variables had the greatest coefficients in the MR model.*

Determining whether these results are statistically significant, can justify the results obtained by the MR model. Results from an ANOVA test showed that an adjusted R^2 of 0.998 was

Variable	Description
x_1	Open
x_2	High
x_3	9MA
x_4	21MA
x_5	Average gain
x_6	Average loss
x_7	Change in closing price
x_8	Positive change in closing price (binary)
x_9	Chikou Span
x_{10}	Positive change in Chikou Span (binary)
x_{11}	Difference between Tenkan and Chikou Span
x_{12}	Positive difference between Tenkan and Lag (binary)
x_{13}	Difference between Kijun and Chikou
x_{14}	Positive difference between Kijun and Lag (binary)
x_{15}	Upper Bollinger Band
x_{16}	Lower Bollinger Band
x_{17}	Change in 1 month implied volatility bid
x_{18}	Positive difference in 1 month implied volatility bid (binary)
x_{19}	Change in 3 Month implied volatility closing price
x_{20}	Positive change in 3 Month implied volatility closing price (binary)
x_{21}	Change in 6 month implied volatility bid price
x_{22}	Change in 9 month implied volatility bid price
x_{23}	Change in 1 month risk reversal bid price
x_{24}	Positive change in 1 month risk reversal bid price (binary)
x_{25}	Change in 3 month risk reversal bid price
x_{26}	Positive change in 3 month risk reversal bid price (binary)
x_{27}	Change in 6 month risk reversal bid price
x_{28}	Positive change in 6 month risk reversal bid price (binary)
x_{29}	Change in 9 month risk reversal bid price
x_{30}	Positive change in 9 month risk reversal bid price (binary)

TABLE 5.1: *A description of all variables used in the MR model.*

obtained. Thus the MR model explains 99.8% of the variance in the daily closing price of the USD/ZAR exchange rate. However, the significance of each variable should also be considered and tested. When the null hypothesis, which states that any variable with a p-value smaller than 0.05 can be rejected by the null hypothesis, is tested for each variable, the results in Table 5.2 are obtained. The 18 variables for which the null hypothesis is accepted should therefore be removed, and the MR model should be fitted on the new variables. The results from removing these variables one at a time and refitting the MR model are shown in Table 5.3. These results show no meaningful impact on performance as the adjusted R^2, F-value, MSE, and accuracy differ with minimal changes. Although removing x_{18} resulted in a higher accuracy of 33.6%, an increase of 3.2%, the remaining measures remained almost unchanged. Therefore, removing variable x_{18} provides no clear advantage as the 3.2% increase in accuracy is negligible. For this reason, the original variables in equation (5.9) were used to fit the MR model for a comparative baseline result. This provides a suboptimal MR model but, for the purpose of this study and to remain within the scope, the optimisation of the MR model is not considered.

Variable	P-value	Hypothesis test
x_1	0.0474	rejected
x_2	0.0	rejected
x_3	0.0	rejected
x_4	0.0	rejected
x_5	0.0004	rejected
x_6	0.8853	accepted
x_7	0.0	rejected
x_8	0.0	rejected
x_9	0.1928	accepted
x_{10}	0.3792	accepted
x_{11}	0.0	rejected
x_{12}	0.9621	accepted
x_{13}	0.0012	rejected
x_{14}	0.5686	accepted
x_{15}	0.1993	accepted
x_{16}	0.0007	rejected
x_{17}	0.0008	rejected
x_{18}	0.1794	accepted
x_{19}	0.5885	accepted
x_{20}	0.1411	accepted
x_{21}	0.138	accepted
x_{22}	0.0846	accepted
x_{23}	0.2501	accepted
x_{24}	0.2011	accepted
x_{25}	0.6787	accepted
x_{26}	0.2577	accepted
x_{27}	0.0461	rejected
x_{28}	0.9376	accepted
x_{29}	0.9928	accepted
x_{30}	0.5613	accepted

TABLE 5.2: *The p-values obtained for all the variables identified by PCA. The result of the null hypothesis, H_0, is shown in last column.*

The change in accuracy as predictions are made, shown in Figure 5.7(a), shows that as predic-

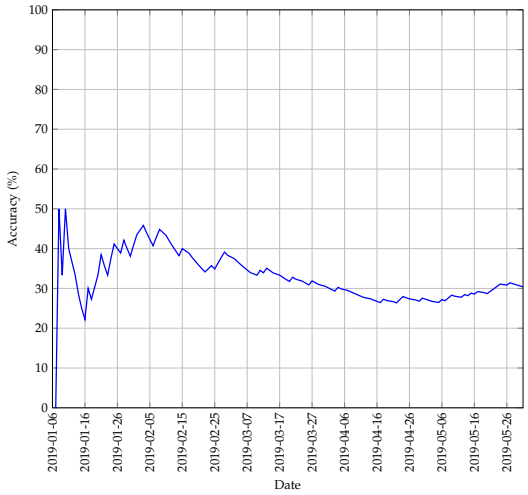
Variable	Adjusted R^2	F-value	MSE	Accuracy (%)
Original	0.9975772	18323.43	0.0041757	30.4
x_6	0.9975790	18969.49	0.0041820	29.6
x_9	0.9975759	18945.09	0.0041868	31.2
x_{10}	0.9975776	18958.53	0.0041770	29.6
x_{12}	0.9975790	18969.76	0.0041738	30.4
x_{14}	0.9975784	18965.06	0.0042069	29.6
x_{15}	0.9975760	18945.80	0.0041801	30.4
x_{18}	0.9975757	18943.53	0.0041935	33.6
x_{19}	0.9975785	18965.53	0.0041794	30.4
x_{20}	0.9975750	18938.25	0.0041702	28.8
x_{21}	0.9975749	18937.75	0.0041656	29.6
x_{22}	0.9975735	18926.49	0.0041462	30.4
x_{23}	0.9975766	18950.53	0.0041601	31.2
x_{24}	0.9975760	18945.99	0.0041771	31.2
x_{25}	0.9975787	18967.29	0.0041698	29.6
x_{26}	0.9975766	18951.13	0.0041731	29.6
x_{28}	0.9975790	18969.70	0.0041734	30.4
x_{29}	0.9975790	18969.79	0.0041755	30.4
x_{30}	0.9975784	18964.88	0.0042024	30.4

TABLE 5.3: *The different ANOVA and performance measures obtained when the variables that failed the hypothesis test were removed from the MR model.*

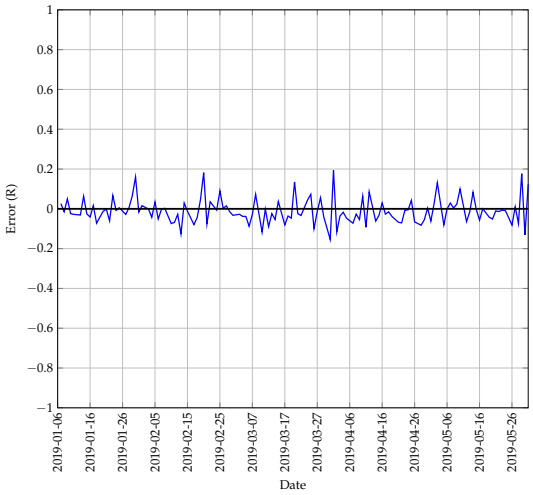
tions are made, the accuracy declines. The initial accuracy jumps to 50% after which it declines to 22.22% and then rises to 45.83%. After this second increase, the accuracy steadily decreases and ends on a final accuracy of 30.4%. The change in error as predictions are made are shown in Figure 5.7b. During the period of 7 January 2019 to 5 February 2019, the error fluctuates around 0 but then increases drastically to 0.16 after which it resides back to fluctuating around an error of 0. This fluctuation seems to repeat itself again from 5 February 2019 to 25 February 2019, 17 March 2019 to 6 April 2019 and 16 May 2019 to 31 May 2019. These fluctuations could be due to volatile price movements in which the MR model could not accommodate the large differences in closing prices. Using these time periods and considering Figure 5.5, an increase in volatile price movement can be seen. This confirms that the MR model has difficulty making predictions when the change in closing price is too large. However, the change in closing price, x_6, is one of the variables that has the greatest influence on predictions. This could suggest that the data used for training did not contain enough volatile changes in price movement or, that on average, x_6 does not have such a great influence on predictions.

However, for the regression model to be accepted, the residuals should have a normal distribution and be indicative of homoscedasticity and no multicollinearity should exist. Figure 5.8 shows the distribution of the residuals, with $\mu = 5.069 \times 10^{-13}$ and $\sigma = 0.0722$, of the MR model described by equation (5.9). By using a Chi-squared test, which states that if $\chi^2 < \chi^2_{crit}$ then the data is normally distributed, with a confidence level of 0.05; the residuals are confirmed to be normally distributed as $\chi^2 = 0.506443 < \chi^2_{crit} = 124.342$. Although the standard deviation differs from the expected value of 1, 74.03% of residuals fell within one standard deviation from the mean and 95.51% of residuals fell within two standard deviations from the mean, confirming the conclusion obtained from the Chi-squared test that the residuals have a normal distribution.

Although the residuals have a normal distribution, it should be indicative of homoscedasticity. Figure 5.9 is a scatter plot of the residuals used to visually examine the assumption of



(a) *The change in accuracy of the MR model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the MR model whilst using a walk-forward validation approach on the test set.*

FIGURE 5.7: *A graphical representation of how the accuracy and error of the MR model changed whilst making predictions on the test set.*

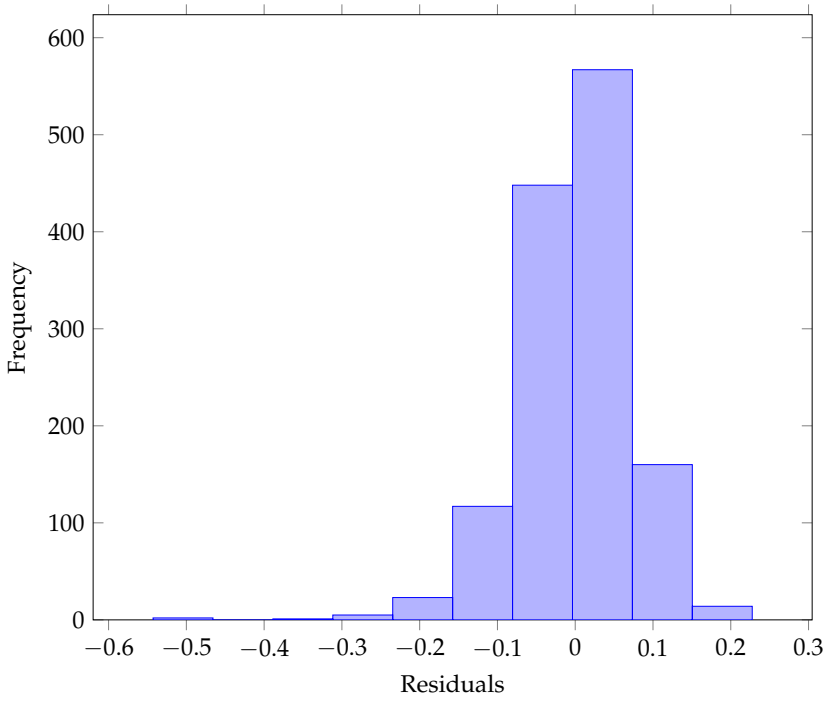


FIGURE 5.8: *The distribution of the residuals of the MR model.*

homoscedasticity. If the residuals do not have a pattern and are randomly scattered around a horizontal line or if the F-value is insignificant, homoscedasticity is satisfied. If a pattern exists, such as the residuals increasing in size, or if the F-value is significant, the assumption is violated and heteroscedasticity is satisfied. The residuals in Figure 5.9 do not form any concise or visible pattern. The F-test is also insignificant as F-value $= 0.0001832 < 0.05$, therefore satisfying the assumption of homoscedasticity. As PCA identified the variables used to fit the MR model, no multicollinearity exists.

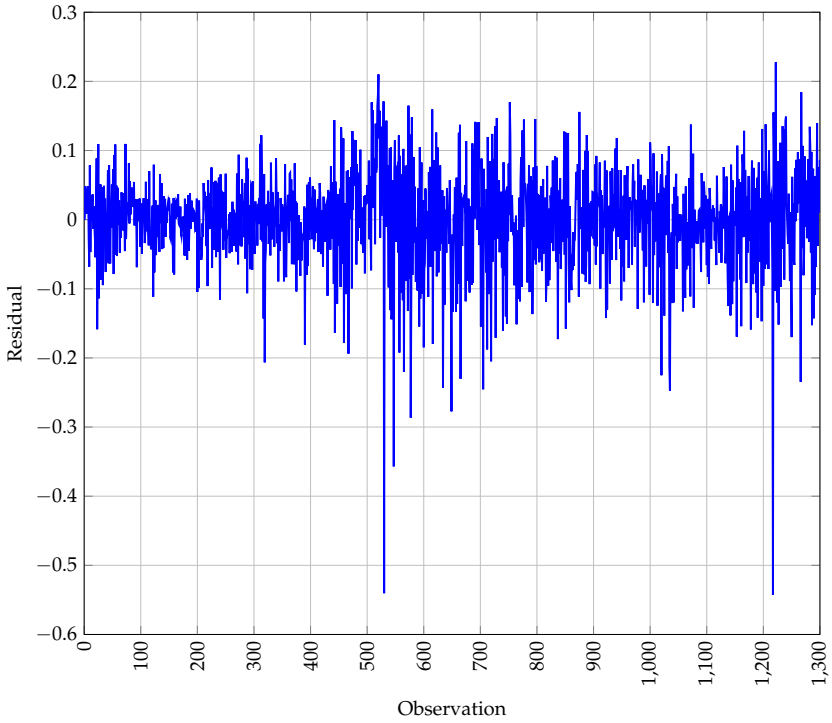


FIGURE 5.9: *A scatter plot of the residuals of the MR model.*

CHAPTER 6

Results for the LSTM model

Contents

6.1 Results for experiments in Group A 47
6.2 Results for experiments in Group B 52

Despite 12 tests being conducted, several tests produced similar results. It was therefore decided that tests which shared similar results (predictions being made, the change in both accuracy and error over time and change in accuracy for different thresholds) would be grouped together and the best result within that group would be reported on. However, all results for all the tests conducted for the LSTM model are shown in Section A.1 of the appendix. The first group of results for the LSTM model are referred to as "Group A" and second group as "Group B". Group A consisted of Experiments 1, 2, 3, 4, 7, 8, 9 and 10; and Group B consisted of Experiments 5, 6, 11 and 12, as defined in Table 3.2.

6.1 Results for experiments in Group A

The grouping of these tests may seem counter-intuitive at first as their input features do not describe similar characteristics. However, their results show distinct similarities. The following results were obtained by test 1, the best performing test, in Group A. Test 1 had the following properties: 50 neurons per layer, 16 layers, 1 day as input, and inputs which can be seen in Table 3.2 and A.73.

Figure 6.1 shows the difference in the best predictions made by the LSTM and MR model, the top-performing statistical model. The LSTM achieved an accuracy of 22.4% with one previous day as input and a MSE of 0.022632, whereas the MR forecast achieved an accuracy of 30.4% and a MSE of 0.004175. The MR therefore outperformed the LSTM model with an 8% increase. However, this is expected when looking at the predictions made in Figure 6.1 as a lag in the LSTM predictions is present. This lag is due to several factors (a scenario seen throughout the results for each model). These factors include experimenting with increased previous timesteps, changing the structure of the model, using different optimisers with different learning rates and using more data to train the model. Due to computing time constraints, optimising these factors that affect network prediction performance could not be tested and improved further.

The change in accuracy, seen in Figure 6.2(a), gives an indication of how volatile the predictions are. The accuracy shown by Figure 6.2(a) is obtained by calculating the accuracy as each

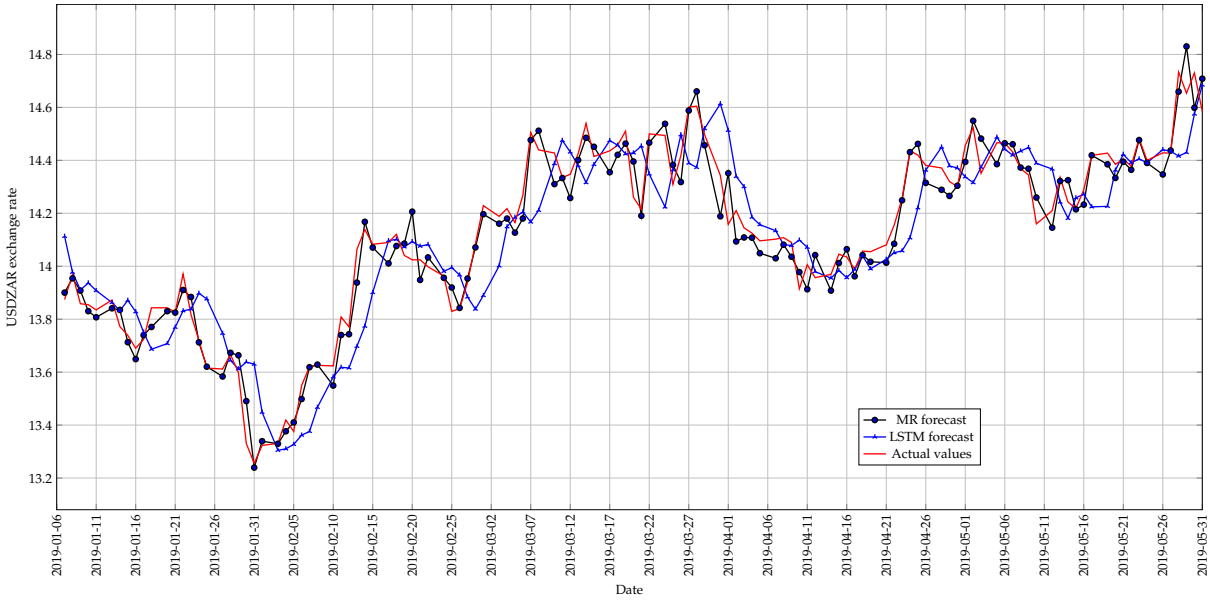
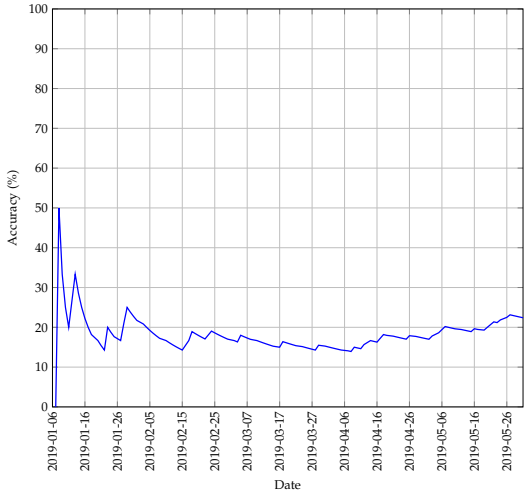
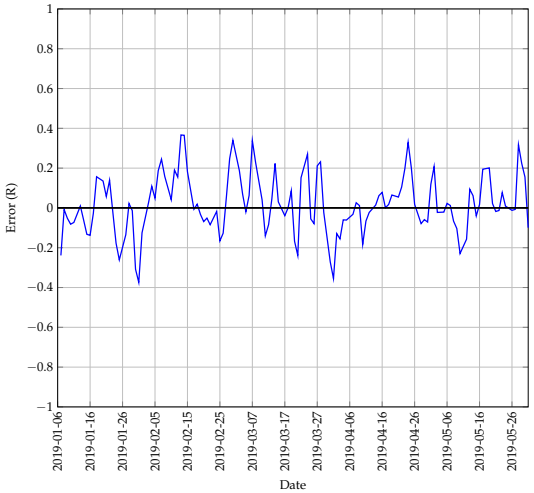


FIGURE 6.1: *The best predictions made by the LSTM model for test 1 with 1 previous day as input using a walk-forward validation approach on the test set.*

prediction is made within the walk-forward validation process (if the first 3 predictions were inaccurate and the fourth prediction is accurate, the accuracy would be 25%). After the 2nd prediction, the accuracy jumps from 0% to 50%. It then proceeds to decrease and fluctuate within the 10%–20% range after which the final accuracy settles to 22.4%. A deeper analysis of the predictions can be done by considering the difference between the predicted and actual values shown in Figure 6.2(b). The error fluctuates between $|0.4|$ with no clear identifiable pattern. This suggests that the features given to the LSTM model are indeed affecting the predictions. However, if a pattern were to exist, perhaps all predictions are consistently negative, it could indicate an error in either the data preprocessing or the LSTM model itself.



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE 6.2: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 1.*

Despite the features given as input to a NN being an important factor to consider when designing a NN, the architecture of the NN should also be considered. Test 1 was used as a baseline for prediction accuracy. A grid search was performed on both the number of neurons in each hidden layer, and the number of hidden layers. The number of neurons and layers to include in the grid search was decided upon after consulting with industry professionals, considering the time required to complete the grid search as well as the consensus in literature that "there is no unique method for fixing the optimum number of neurons in hidden layer[s] for a particular problem" [22]. Figure 6.3 illustrates the impact that different architectures have by showing the accuracy obtained for different architectures of the LSTM model. An element that immediately stands out from Figure 6.3 is the clear separation between adequate and poor prediction accuracies. The separation was not expected and proved to be a vital finding that eased tuning network parameters for the succeeding tests. The separation also highlights the sensitive nature of neural networks as adding one additional hidden layer to an existing network can drastically impact the network's predictive performance.

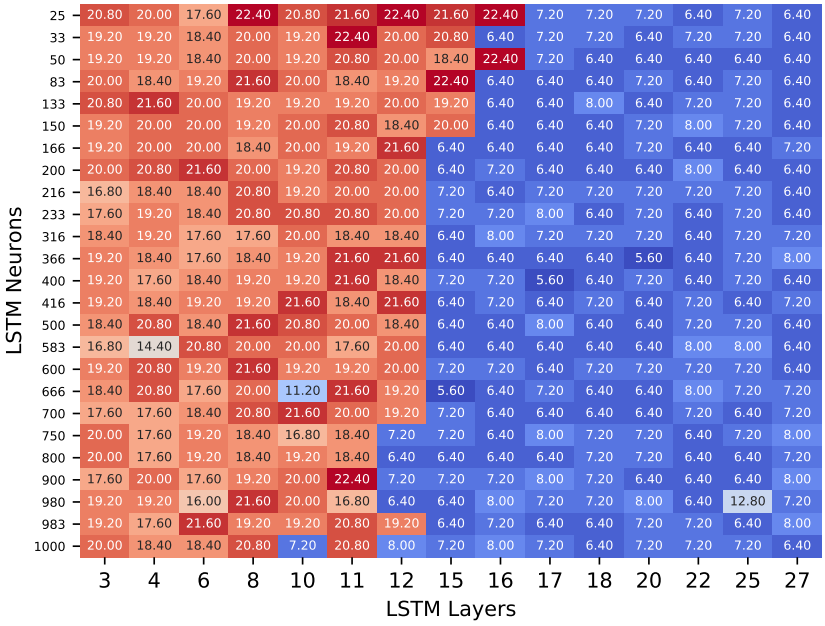


FIGURE 6.3: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 1 with 1 previous day as input.*

An interesting conclusion can be drawn when the threshold used to calculate the accuracy is increased for the different models. Figure 6.4 depicts this incremental change of the accuracy threshold and the influence it has on the overall prediction accuracy. The increase in overall accuracy could initially be described as linear. However, the accuracy shown in Table 6.1 shows that this is not the case. This becomes evident when the change in accuracy is transformed to a ratio. As each threshold is a multiple of 0.025, accuracy is expected to increase by the ratios $2, 3$ and 4 respectively. However, the observed ratios for the naïve forecast are actually $1.676, 2.236$, and 2.618 respectively with similar ratios obtained for the SES and LSTM model. However, the observed ratios for the MR model are 1.957, 2.605, and 2.973 respectively. Although the MR ratios are slightly closer to the expected ratios of 2,3 and 4, all the model ratios follow the same trend. This follows a pattern similar to an increasing function with a decreasing rate of change. This is expected as the threshold and prediction accuracy are not directly proportional. Increasing the threshold would increase the accuracy, as seen in Figure 6.4, as it considers new correct predictions. However, the accuracy would not double as the number of correct predictions

did not double — only the addition of several, if any, predictions are now considered.

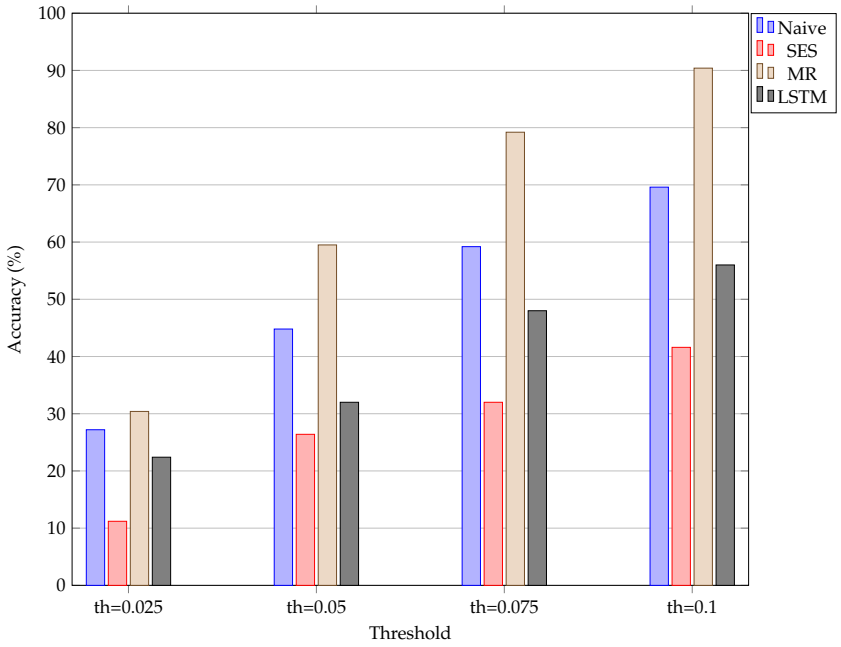


FIGURE 6.4: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	22.4	32.0	48.0	56.0

TABLE 6.1: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1.*

Although the accuracy for all models and tests is determined using the threshold, another interesting factor to consider is the direction of the actual price movement versus the movement of the predicted values. This changes the problem from regression, predicting the closing price, to classification, predicting the up or down class. This study does not require building nor solving a classification problem (a classification problem would require a different modelling structure). For the purpose of this study, the classification results are shown as additional results that could potentially lead to further research.

The accuracy obtained for predicting the correct direction of price movement for the different models is shown in Figure 6.5 with the actual values shown in Table 6.2. As expected, the MR model again outperforms the LSTM with a classification accuracy of 70.161%, an increase of 24.193% from the LSTM model. The accuracy obtained by reconsidering the predictions as classifications is substantially higher than the accuracy obtained when the default regression problem with a R0.025 threshold is considered. This is expected as the only factor taken into consideration is whether the predicted closing price follows the same direction as the actual closing price.

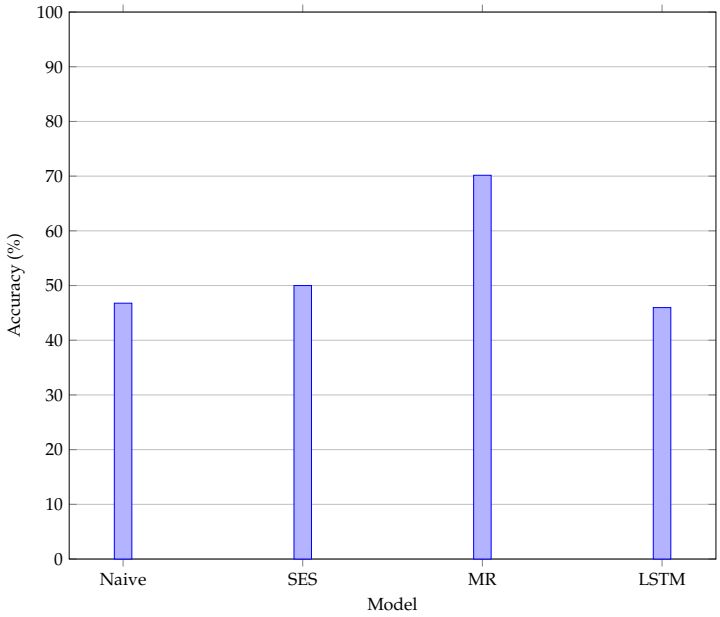


FIGURE 6.5: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
LSTM	45.968

TABLE 6.2: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1.*

6.2 Results for experiments in Group B

The results that follow were obtained for test 5, the best performing test, in Group B. Test 5 had the following properties: 31 neurons per layer, 11 layers, 2 days as input, and inputs which can be seen in Table 3.2 and A.73. Figure 6.6 illustrates the difference between the predictions made by the LSTM and MR model. Similarly to the predictions in Group A, a lag in the predictions is present for the LSTM model. However, the distinguishing factor that separates the predictions in Group A and B is the scale of the predictions. With Group B, the initial predictions are all shifted upwards (average error increased from 0.082163 to 0.249754) with varying factors with some predictions being shifted upwards by almost R0.20 whereas others are shifted by less than R0.1. After the initial predictions the remaining predictions, with the exception of a few random predictions, also seems to be scaled by some factor. As expected, the scaling of the predictions decreased the prediction accuracy of the LSTM model, obtaining an accuracy of 17.6% with 2 previous days as input. This is a decrease of 4.8% when compared to the best prediction accuracy in Group A. The LSTM also achieved an MSE of 0.036063, an increase in error of 0.013431.

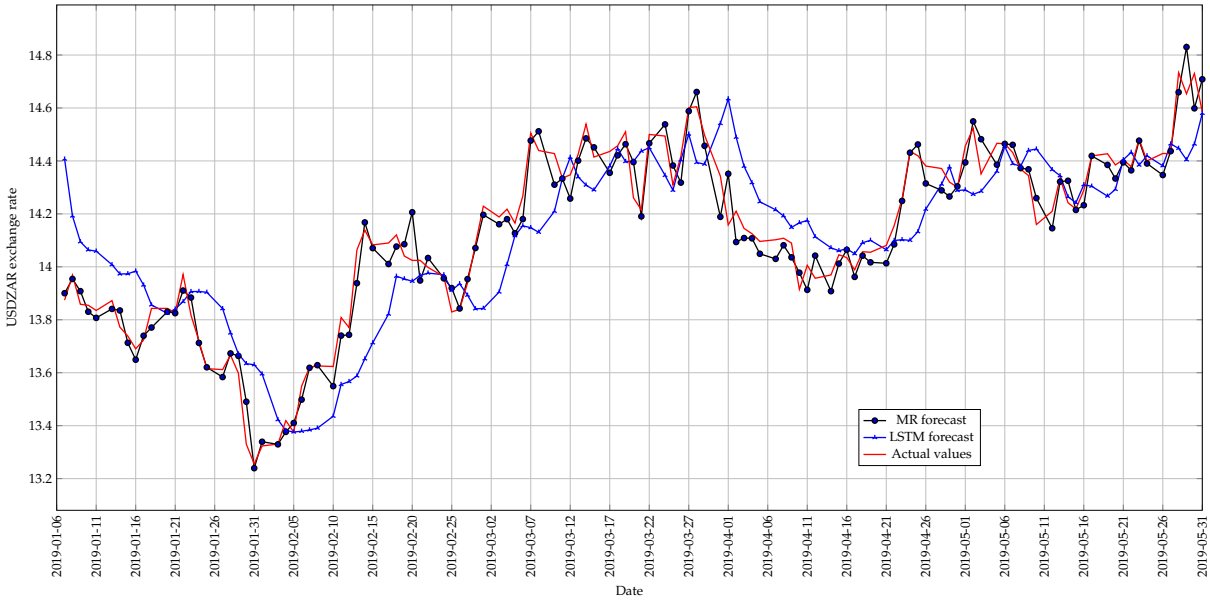
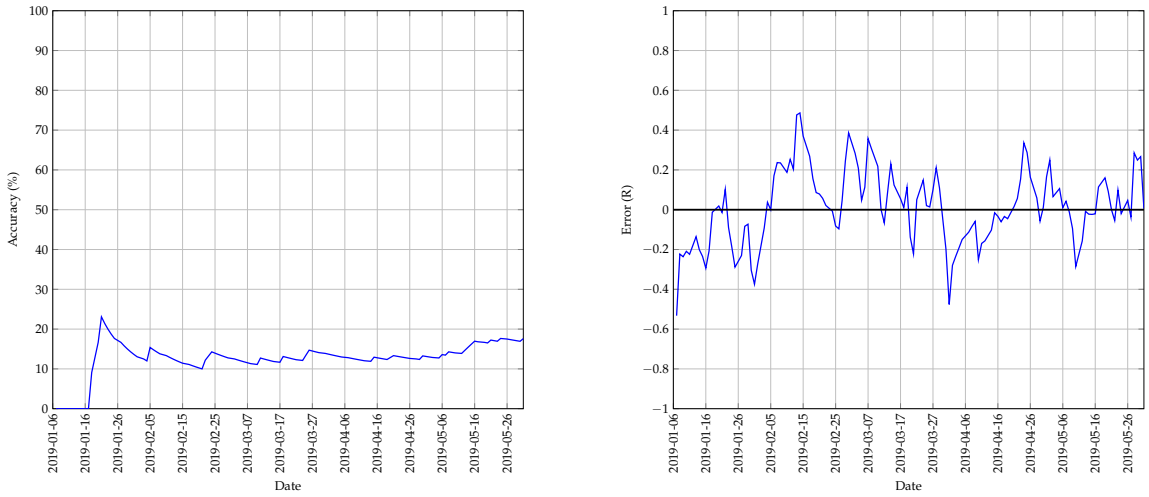


FIGURE 6.6: *The best predictions made by the LSTM model for test 5 with 1 previous day as input using a walk-forward validation approach on the test set.*

Another factor that separates Group A and B is the change in accuracy and error as the predictions are being made. Where Group A saw an initial increase followed by a strong decline and levelling out around 20% for the change in accuracy, Group B saw a 0% accuracy, seen in Figure 6.7(a), for the initial predictions which then followed a spike in accuracy to less than half of that seen in Group A. The change in error for Group B, seen in Figure 6.7(b), also saw errors greater than |0.4| with most of the errors fluctuating further away from the R0 error line when compared to those seen in Group A.

Figure 6.8 shows the network architectures tested for test 5. Since test 1 in Group A formed the baseline test from which network architecture would be tested, the tests conducted for test 5 were on a subset of those tested in test 1. Once a higher accuracy was obtained, a greedy local search was conducted around the network architecture responsible for the higher accuracy. Unlike the clean separation seen in Figure 6.3, the accuracy obtained for test 5 seems more erratic with random tests yielding poor accuracy. This is seen when looking at 11 hidden layers



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE 6.7: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 5.*

and 31 neurons in each layer. When the number of neurons is increased or decreased by 1, the accuracy drops by 10.4% and 9.2% respectively. This large fluctuation in prediction accuracy when experimenting with network architecture makes experimentation difficult and counter-intuitive. The assumption of following a greedy search does not, in this case, yield the expected results.

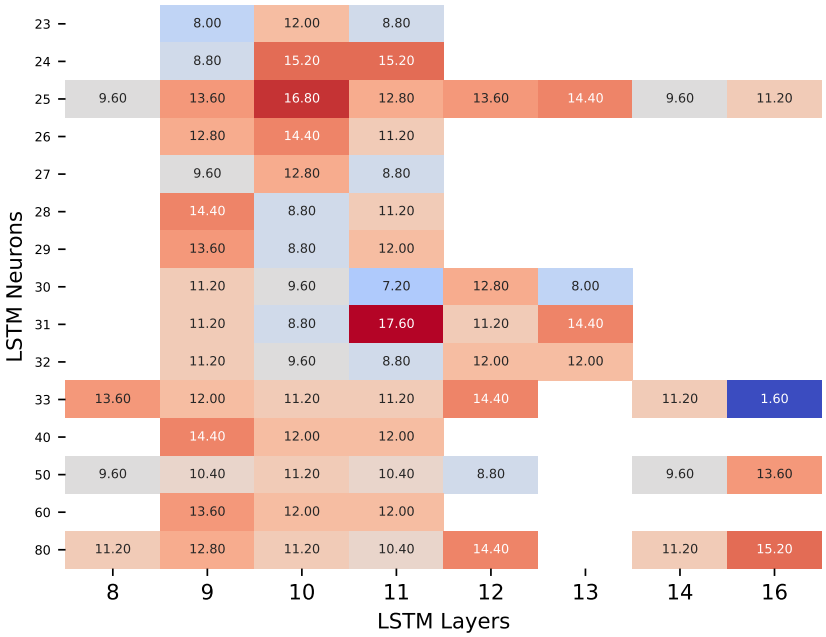


FIGURE 6.8: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 5 with 2 previous day as input.*

Observing the change in accuracy as the threshold is incrementally increased provides another measure of comparison between Groups A and B. These changes are visualised in Figure 6.9

and tabulated in Table 6.3. For Group A it was seen that when the change in accuracy is transformed to a ratio, the increase in accuracy did not follow the expected trend but rather that of an increasing function with a decreasing rate of change. The same result is seen with Group B. The actual ratios obtained by the LSTM model were 1.409, 1.955, and 2.545 where the expected ratios are 2, 3 and 4. These ratios are substantially lower than those obtained in Group A with the first and second ratio being lower than 2. This illustrates both the poor performance of the tests run in Group B as well as the increasing function with a decreasing rate of change trend seen in Group A.

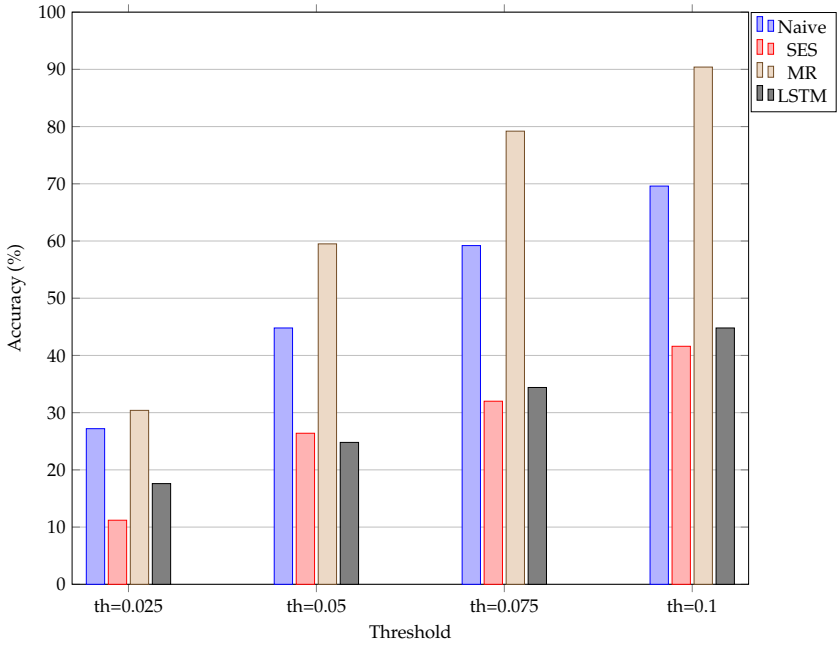


FIGURE 6.9: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM (A)	22.4	32.0	48.0	56.0
LSTM (B)	17.6	24.8	34.4	44.8

TABLE 6.3: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5.*

The last factor to compare results between Groups A and B is the accuracy obtained when a classification measure is taken. Test 5 obtained the highest accuracy of 49.194% when only the direction of price movement of the predicted values is considered. The expected result would be that Group A, with a higher prediction accuracy obtains a higher classification accuracy, but as seen in Figure 6.10 and Table 6.4 this is not the case. This unexpected result of test 5 outperforming Group A by 3.226% is due to predictions, despite not being within the accuracy threshold, moving more frequently in the correct direction. This is clear when considering a prediction that has an error of R0.5, but the prediction is in the correct direction of actual price

movement.

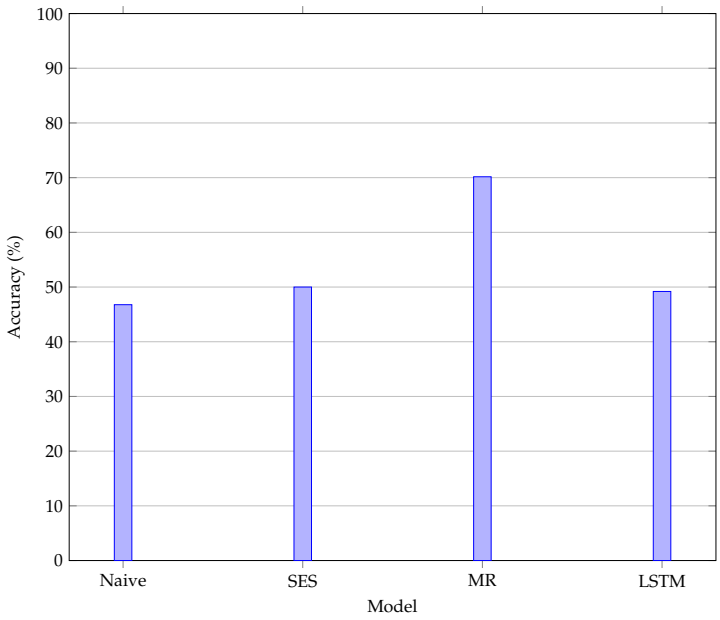


FIGURE 6.10: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
LSTM (Group A)	45.968
LSTM (Group B)	49.194

TABLE 6.4: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5.*

CHAPTER 7

Results for the CNN-LSTM model

Contents

7.1 Results for experiments in Group A 57
7.2 Results for experiments in Group B 61

The results of the CNN-LSTM model were grouped according to tests that obtained similar results (same grouping measures as the LSTM model). The CNN-LSTM groups will be referred to as "Group A" and "Group B" where the best performing test for each group will be reported on. All the results for all the tests conducted can be seen in Section A.2 of the appendix. Group A consisted of Experiments 1, 2, 3, 4, 7, 8, and 10; and Group B consisted of Experiments 5, 6, 9, 11 and 12, as defined in Table 3.2.

7.1 Results for experiments in Group A

The results discussed here were obtained for test 1, the best performing test in Group A. Test 1 had the following properties: 105 CNN neurons per layer, 750 LSTM neurons per layer, 12 CNN layers, 4 LSTM layers, 2 days as input, and inputs which can be seen in Table 3.2 and A.73. The best predictions for Group A were obtained by test 1 with an accuracy of 24.8% and a MSE of 0.024472 using 2 previous days as input. These predictions are shown in Figure 7.1 and resemble those seen in Group A of the LSTM model. The MR model outperformed the CNN-LSTM with an accuracy of 30.4% and a MSE of 0.004175. Since the CNN-LSTM predictions contain a lag (explained in Section 6.1), this 5.2% difference in accuracy is expected.

Considering the change in accuracy and error as predictions are being made, seen in Figures 7.2(a) and 7.2(b), can provide an insight into how predictions are being made. The change in accuracy starts at 0% initially and then jumps to 16.66% after 6 predictions. It then proceeds to decline to 7.14%, fluctuate around 20% and then finally ends on 24.8%. In comparison to LSTM Group A, predictions 35–79 show similarities with the change in accuracy seen in the CNN-LSTM model. Both models see a decline in accuracy in this range followed by an increase in accuracy as the final predictions are being made. This could indicate a period of uncertainty

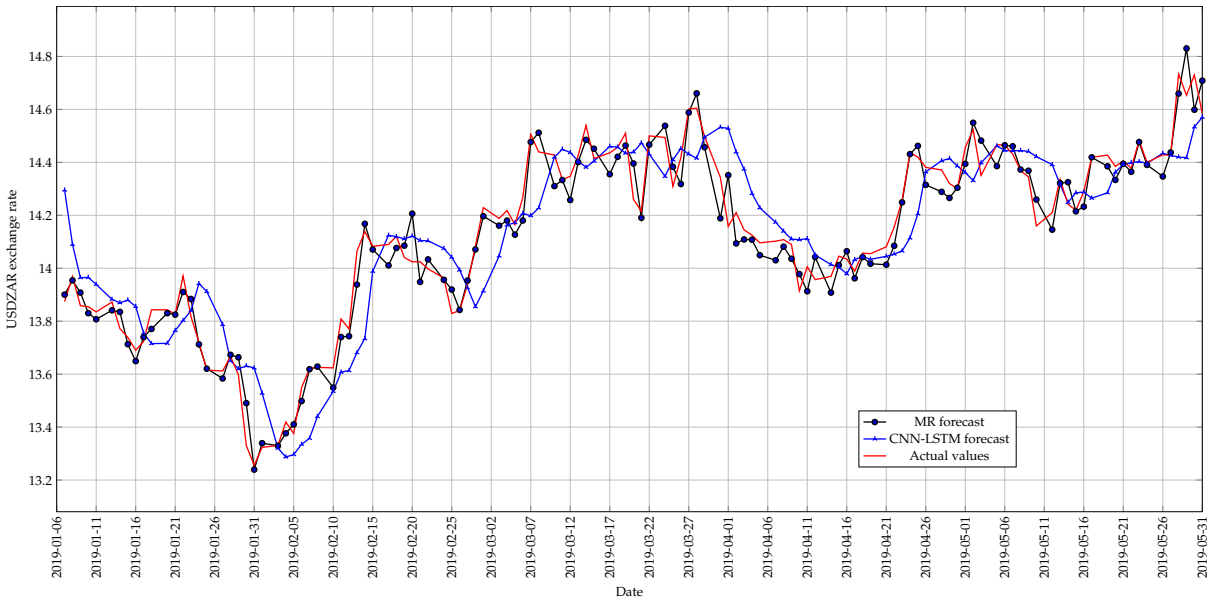
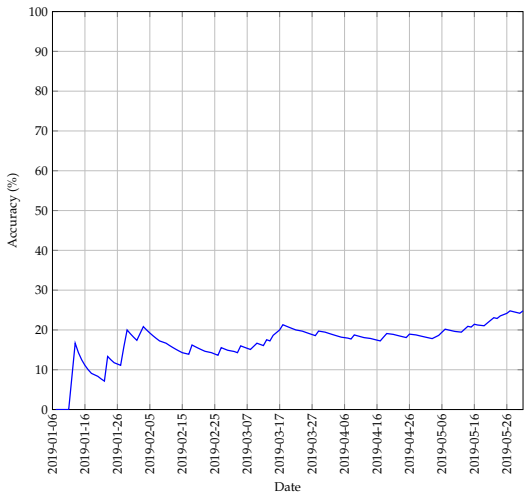
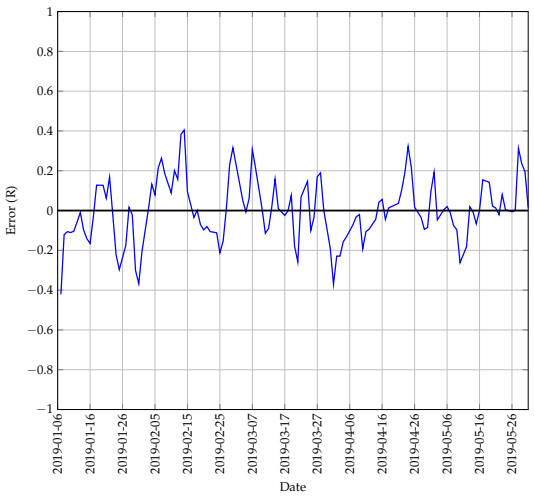


FIGURE 7.1: *The best predictions made by the CNN-LSTM model for test 1 with 2 previous day as input using a walk-forward validation approach on the test set.*

within the predictions where the input features do not provide enough relevant information for accurate predictions to be made. The change in error fluctuates between an error of $|0.4|$ with no clear identifiable pattern. This suggests that the input features are indeed affecting the predictions and that the CNN-LSTM model has not identified a "pattern" to make predictions.



(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

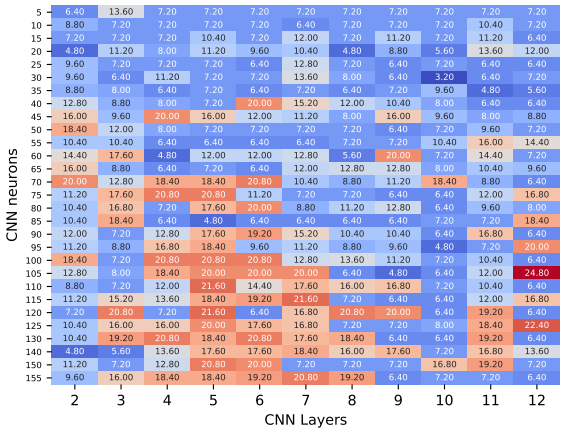


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

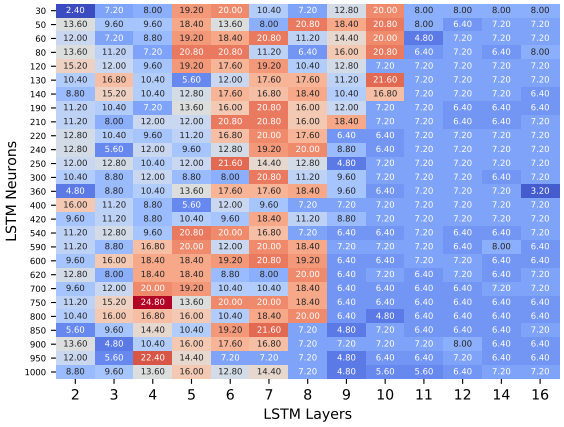
FIGURE 7.2: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 1.*

Figures 7.3(a) and 7.3(b) show the network architectures experimented with for test 1 of the CNN-LSTM model. As the CNN-LSTM model has both a CNN and LSTM component, the network architectures had to be tested separately. Figure 7.3(a) shows the network architectures tested for the CNN and Figure 7.3(b) shows the network architectures tested for the LSTM. As

with test 1 in Group A of the LSTM model, a grid search was performed on both the number of neurons and layers for both the CNN and LSTM components. The unpredictable nature of experimenting with neural network architectures is easily recognised within Figures 7.3(a) and 7.3(b) as no clear separation in accuracy exist. However, the CNN architecture seems to produce adequate prediction accuracies in the range of 90 neurons and 4–7 layers with an outlier, which happens to be the best prediction accuracy, at 105 neurons and 12 layers. This illustrates the unpredictable and counter-intuitive experience when working with neural networks. It would be expected that similar architectures within close proximity of the best performing architecture would obtain similar results. A similar experience is obtained for the LSTM component where no clear identifiable separation exists. However, a possible linear relationship between the number of neurons and layers could exist when looking at the adequate orange prediction accuracies obtained in Figure 7.3(b). This possible linear relationship also contains the best prediction accuracy, strengthening the possibility of a linear relationship.



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 2 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 2 previous days as input.*

FIGURE 7.3: *A graphical representation of the influence that different architectures have on prediction accuracy for test 1 of the CNN-LSTM model.*

Although the accuracy threshold was set at R0.025, an interesting factor to consider is how the accuracy changes when the threshold is increased incrementally. As with the LSTM model, the threshold was increased incrementally by R0.025. The accuracy obtained when increasing the threshold is shown in Figure 7.4 and tabulated in Table 7.1. It would be expected that the accuracy increases by the ratios 2, 3 and 4 respectively. However, this is not the case as the accuracy for the CNN-LSTM and the statistical models are substantially lower than the expected ratios. The CNN-LSTM accuracy changed by 1.322, 1.581 and 2.097 respectively. These ratios are similar to those seen in the LSTM model and follow the same increasing function with a decreasing rate of change.

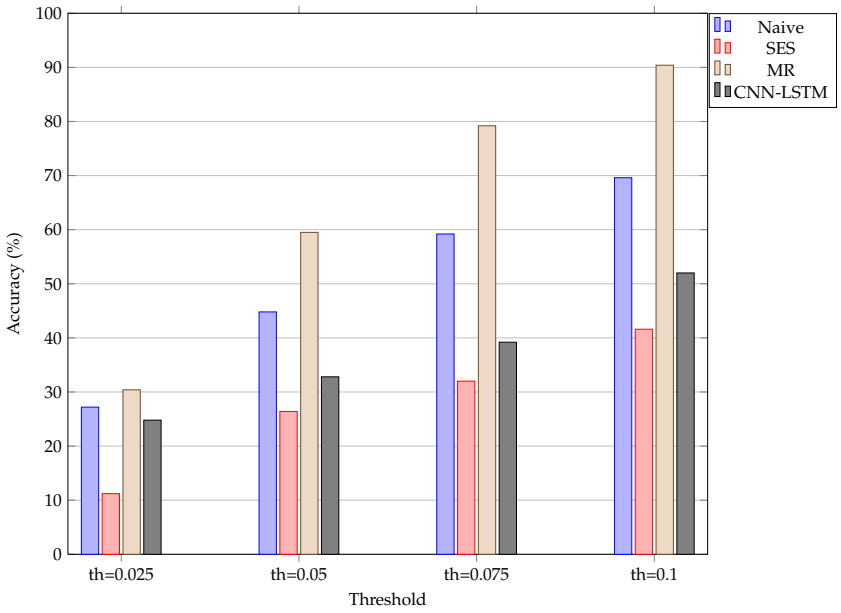


FIGURE 7.4: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	24.8	32.8	39.2	52.0

TABLE 7.1: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1.*

Considering the accuracy of the CNN-LSTM when predicting the direction of price movement can assist in decision making. Test 1 obtained the best accuracy of 45.968%, shown in Figure 7.5 and tabulated in Table 7.2. However, the MR model outperformed the CNN-LSTM by 24.193%, suggesting that, for practical use, the MR model should be used.

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	45.968

TABLE 7.2: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1.*

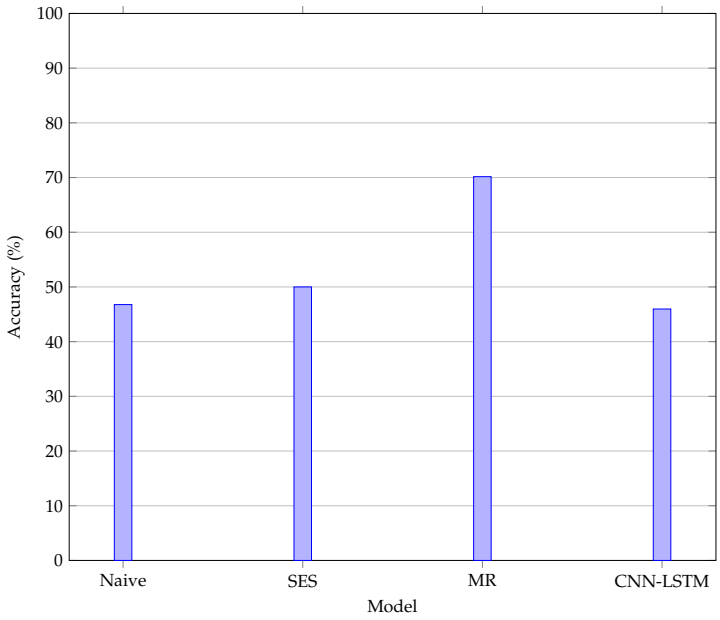


FIGURE 7.5: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1.*

7.2 Results for experiments in Group B

The results that follow were obtained for test 5, the best performing test in Group B. Test 5 had the following properties: 121 CNN neurons per layer, 860 LSTM neurons per layer, 14 CNN layers, 5 LSTM layers, 3 days as input, and inputs which can be seen in Table 3.2 and A.73. Test 5 obtained the best predictions in Group B with an accuracy of 22.4% and MSE of 0.029406 using 3 previous days as input. The predictions of Group B resemble those of Group A but with the first 50 predictions varying. It seems as if these first 50 predictions have been scaled by some factor as well as being shifted in some cases. The biggest difference occurs around predictions made on 5 February 2019 and 20 February 2019. The predictions made during this time seem to be smoothed out and shifted downwards. This causes the error to increase during this period and create uncertainty about the input features. A possible explanation could be that the input features do not capture enough relevant information during this time and that the CNN-LSTM model has to "guess" what the predictions are. It is also seen that the MR model outperforms the CNN-LSTM model with an accuracy of 30.4%, an increase of 8 percent, and MSE of 0.004175, a decrease of 0.025231.

This decline in prediction accuracy during the period of 5–20 March 2019 is visible in Figure 7.7(a), which shows the change in accuracy as predictions are made. The accuracy drops from 12.5% to 8.1%. Although a 4.4% decline in accuracy is not very big, it is enough to cause all predictions in that period to be incorrect. However, accuracy increased after the decline and fluctuated below 20% where it finally ended on 22.4%. In comparison to the predictions in Group A, the 2.4% difference in accuracy of Group B is caused by the incorrect predictions during the aforementioned period. The remaining predictions were all observed to be very similar to those seen in Group A. The change in error shown in Figure 7.7(b) confirms the period of lower accuracy as the error spikes above the $|0.4|$ range that was seen in Group A. The error for this period steadily increases above an error of 0.4 after which it decreases and remains within the expected $|0.4|$ range.

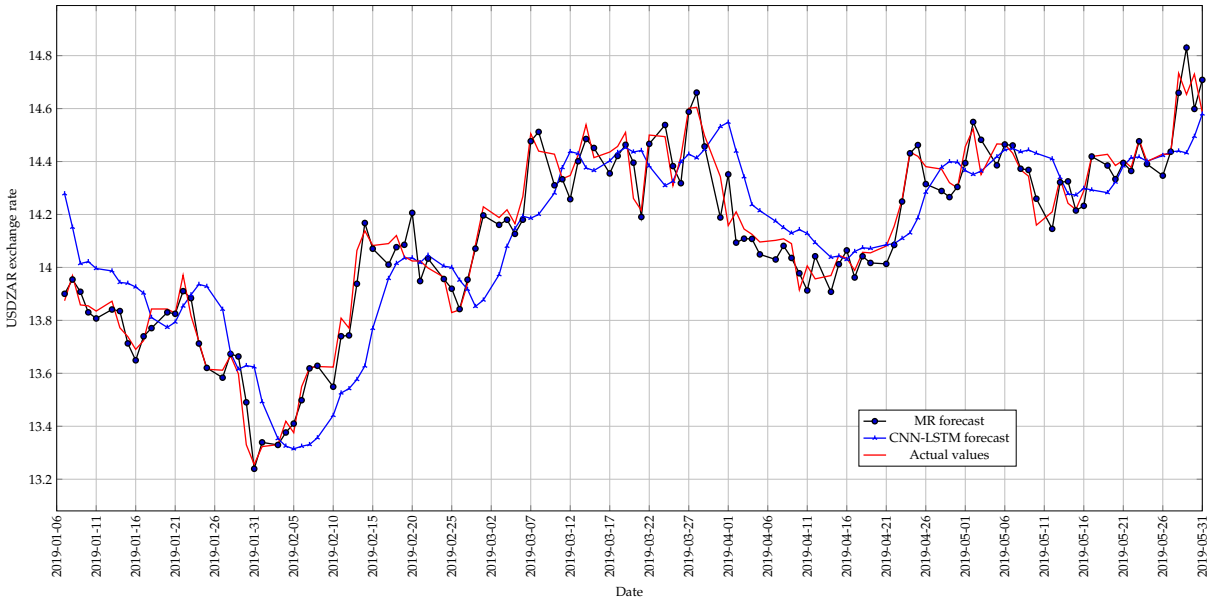
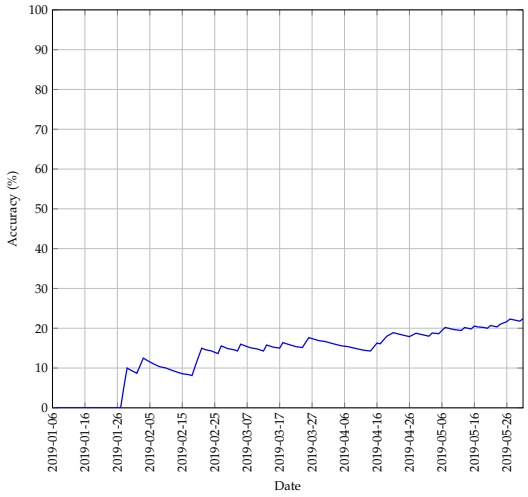
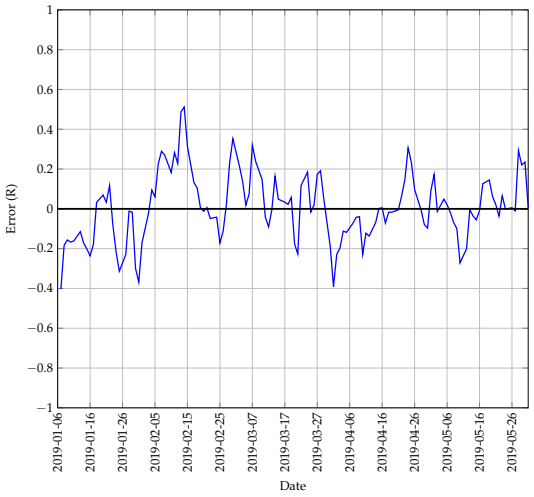


FIGURE 7.6: *The best predictions made by the CNN-LSTM model for test 5 with 3 previous days as input using a walk-forward validation approach on the test set.*



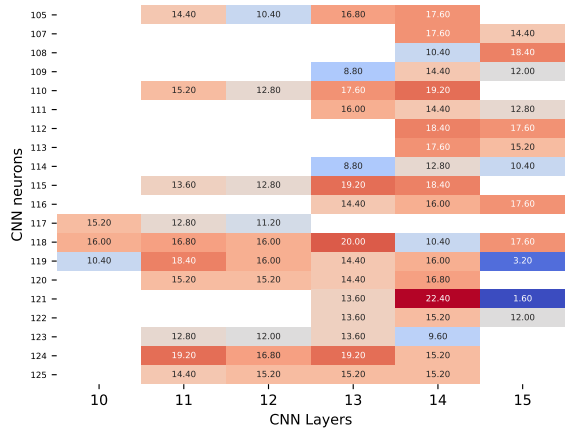
(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



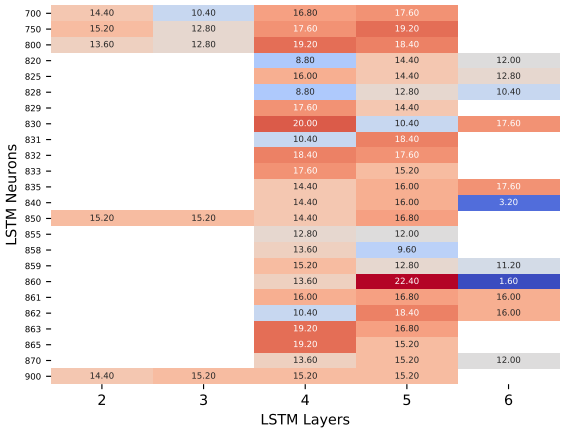
(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE 7.7: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 5.*

Figures 7.8(a) and 7.8(b) show the influence that network architecture has on prediction accuracy. As with Group B in the LSTM model, the architectures tested were on a subset of the best performing architectures in test 1. Once a higher accuracy is obtained, a greedy search would be done around that specific architecture. Figure 7.8(a) shows the accuracy obtained for the CNN component and, as expected, an element of unpredictability exists in the relationship between network architecture and accuracy. If the best performing architecture, with 14 layers and 121 neurons, is increased by one layer, the accuracy drops by 20.8%. This result provides insight into the sensitive nature of neural networks. In Figure 7.8(b) the same result is seen where the accuracy also drops by 20.8% if the number of LSTM layers is increased by 1. In both Figures 7.8(a) and 7.8(b) no clear identifiable pattern exists. This is somewhat expected as only a subset of the network architecture was tested.



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 3 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 3 previous days as input.*

FIGURE 7.8: *A graphical representation of the influence that different architectures have on prediction accuracy for test 5 of the CNN-LSTM model.*

If the accuracy threshold is changed, a pattern amongst the results has emerged. It is seen that when the threshold is increased incrementally by R0.025, the increase in accuracy ratio does not follow the expected trend. The expected result would be that the accuracy increases by the ratios 2, 3, and 4 respectively. However, this is not the case as the CNN-LSTM model increased by the ratios 1.429, 1.893, and 2.179 respectively. The accuracy obtained for the statistical models as well as the CNN-LSTM model is shown in Figure 7.9 and tabulated in Table 7.3 and all follow similar accuracy ratios as those stated above. The increase in ratios has been identified to follow that of an increasing function with a decreasing rate of change.

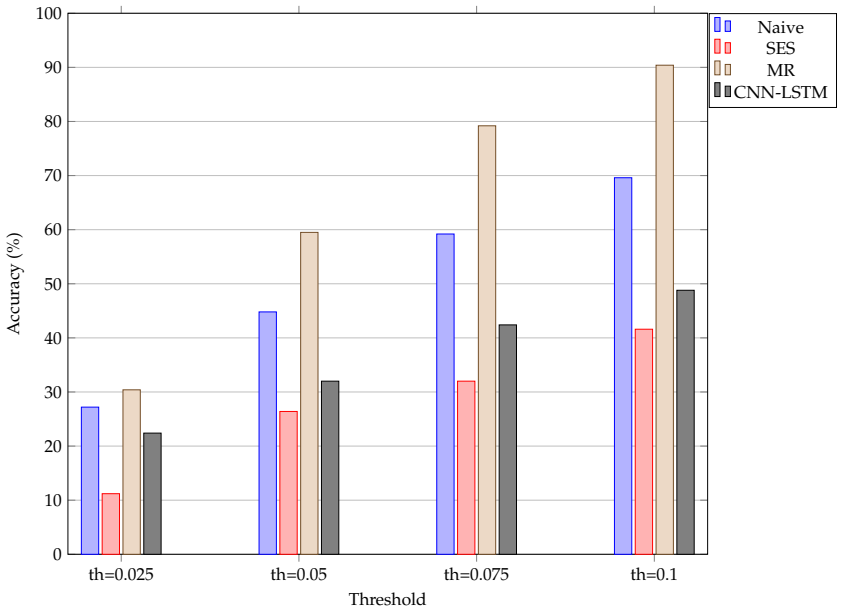


FIGURE 7.9: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	22.4	32.0	42.4	48.8

TABLE 7.3: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5.*

The last factor to consider is when the predictions are used in a classification scenario. Figure 7.10 illustrates the accuracy obtained by the different models, which is tabulated in Table 7.4, when using their predictions to classify price movement. Test 5 obtained the best accuracy of 49.194%, an increase of 3.229% from Group A. An accuracy of 49.194% would therefore not be beneficial in a practical scenario and would not assist in decision making.

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	49.194

TABLE 7.4: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5.*

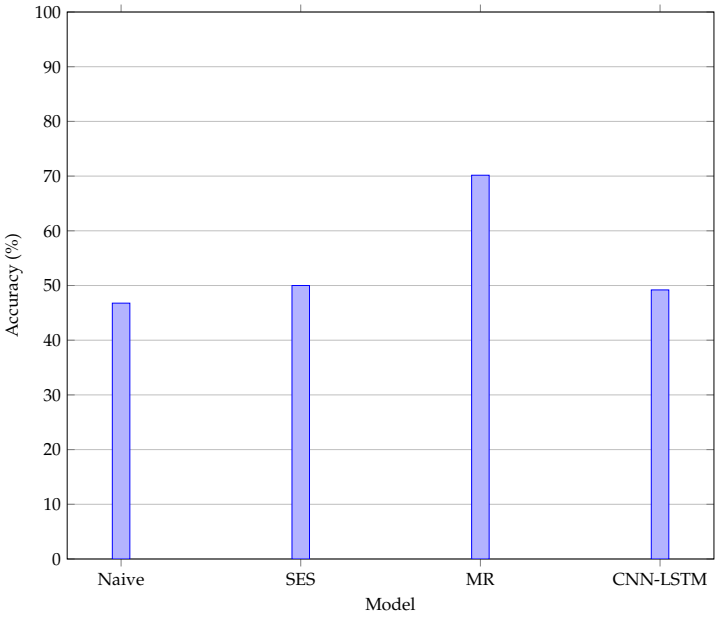


FIGURE 7.10: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5.*

CHAPTER 8

Results for the Multi-head CNN-LSTM network

Contents

8.1 Results for experiments in Group A 67
8.2 Results for experiments in Group B 71

The results obtained from the Multi-head CNN-LSTM model were grouped according to results that were similar. The groups for the Multi-head CNN-LSTM model will be referred to as "Group A" and Group B" where the best performing test within each group will be reported on. All results obtained for each test can be found in Section A.3 of the appendix. Group A consisted of Experiments 1, 2, 3, 4, 7, 8, 9 and 10; and Group B consisted of Experiments 5, 6, 11 and 10, as defined in Table 3.2.

8.1 Results for experiments in Group A

The results that follow were obtained for test 7, the best performing test in Group A. Test 7 had the following properties: 57 CNN neurons per layer, 57 LSTM neurons per layer, 2 CNN layers, 7 LSTM layers, 1 day as input, and inputs which can be seen in Table 3.2 and A.73. Test 7 obtained the best predictions with an accuracy of 25.6% and MSE of 0.024824, using 1 previous day as input. These predictions are shown in Figure 8.1 and resemble those seen in Group A of the LSTM and CNN-LSTM models. The MR model outperforms the Multi-head model with an accuracy of 30.4%, an increase of 4.8%, and MSE of 0.004175, a decrease of 0.020649.

The change in accuracy, shown in Figure 8.2(a), is similar to that seen in Group A of the LSTM model. However, an initial spike to 50% followed by fluctuations in the range of 20%–30%, indicates that the initial predictions of Group A are stable[1] in comparison to those made in Group A of the LSTM and CNN-LSTM models. This is also seen in Figure 8.2(b) which shows the change in error as predictions are made. After the first prediction, the error remains in a range of $|0.2|$ after which it increases and fluctuates between a range of $|0.4|$. A possible

[1] "[S]table" in this context refers to predictions that share similar errors. If predictions had major differences in error, the change in error would fluctuate between a greater range.

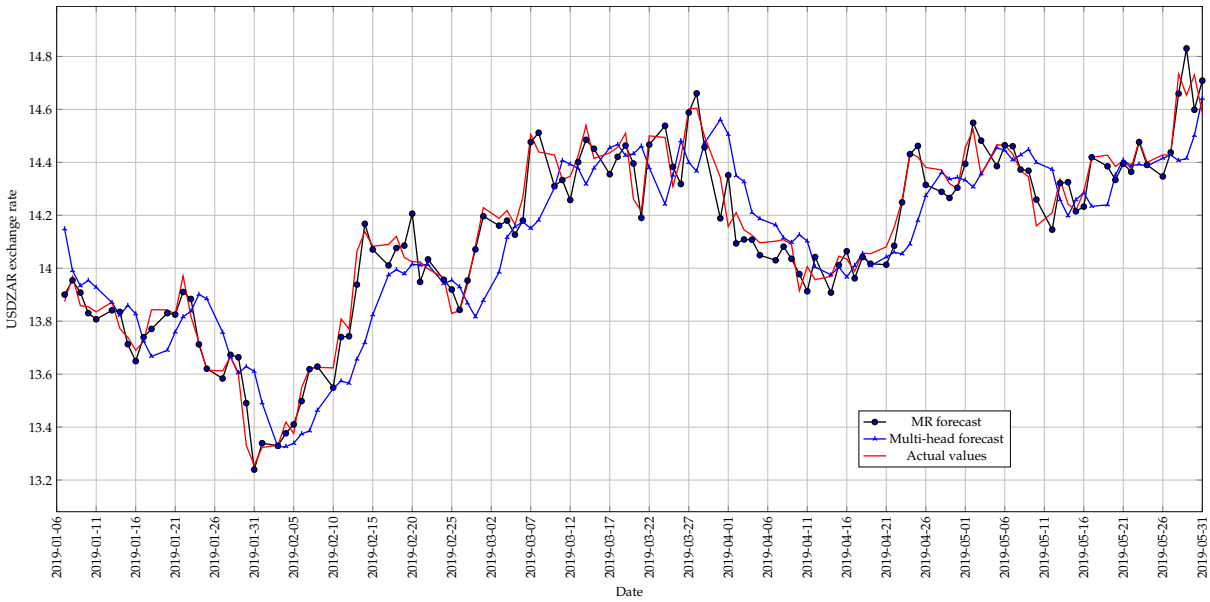
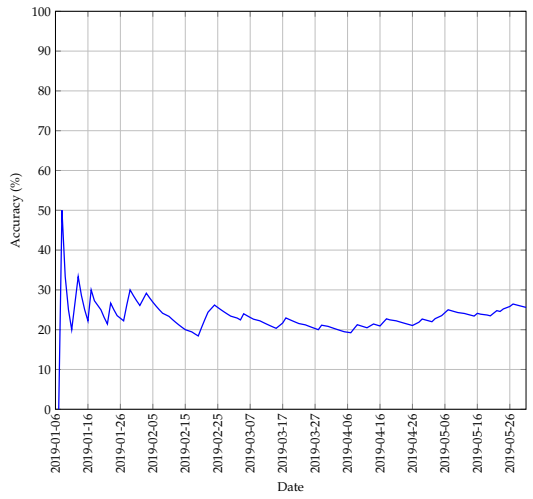
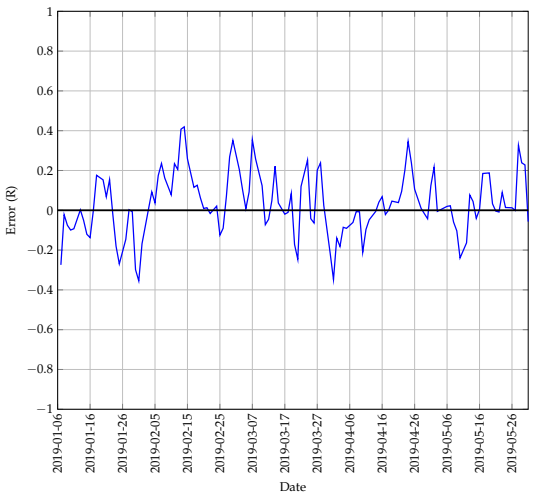


FIGURE 8.1: *The best predictions made by the Multi-head model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set.*

explanation for "stable" initial predictions is that the input features capture more relevant information that the Mutli-head CNN-LSTM can utilise for predictions.



(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

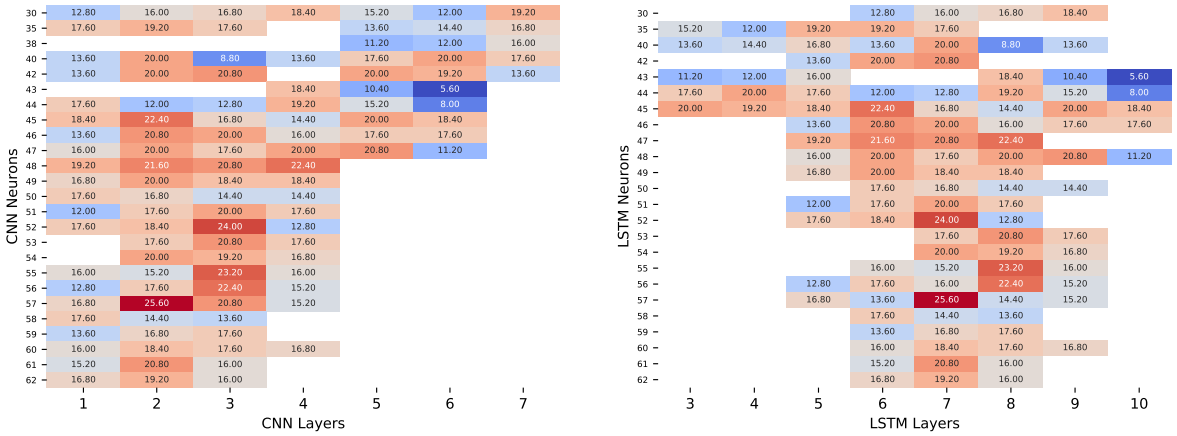


(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE 8.2: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 7.*

The importance of network architecture, and its influence on prediction accuracy is shown in Figures 8.3(a) and 8.3(b). The gaps in tested network architectures for test 7 are a result of testing a subset of the best performing network architectures of test 1. Both Figures 8.3(a) and 8.3(b) show an element of randomness for accuracy obtained with specific architectures. In Figure 8.3(a) the highest accuracy was obtained with 2 CNN layers and 57 CNN neurons. However, with an increase or decrease of 1 neuron, the accuracy drops by as much as 11.2%. The

same result is seen in Figure 8.3(b), with the highest accuracy obtained with 57 LSTM neurons and 7 LSTM layers, where an increase in 1 neuron caused a drop of 12%. This counter-intuitive result is both expected and unexpected. As seen with the results from the other models and knowing that neural networks are sensitive to parameter changes, the heightened sensitivity of the Multi-head CNN-LSTM is unexpected. A small change in accuracy would be expected when changing the architecture by 1 neuron, but not a change as big as 12%. There is also no clear indication of a relationship between prediction accuracy and network architecture when looking at both Figures 8.3(a) and 8.3(b). However, a possible relationship could exist in Figure 8.3(a) with CNN neurons in the range of 48–57 and CNN layers in the range of 1–4.



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE 8.3: *A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the Multi-head model.*

Figure 8.4 and Table 8.1 show how the accuracy changes when the accuracy threshold is incrementally increased by R0.025. Although the accuracy increases with each increasing threshold, the accuracy does not increase as expected. The expected increase in accuracy for each threshold would have the ratios of 2, 3 and 4 respectively. However, the actual ratios that were obtained for the Multi-head CNN-LSTM were 1.406, 1.719, and 2.156 respectively. This increase follows the same trend as seen with the LSTM and CNN-LSTM model where the increase in accuracy follows an increasing function with a decreasing rate of change.

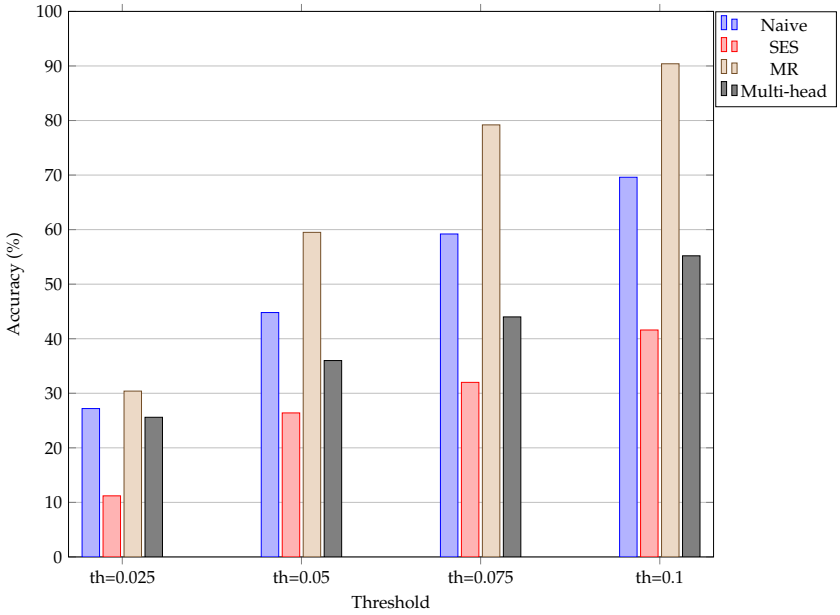


FIGURE 8.4: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	25.6	36.0	44.0	55.2

TABLE 8.1: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7.*

Changing the accuracy threshold provides a method of identifying a threshold that obtains an acceptable accuracy. However, taking a classification approach to the predictions provides a method of identifying what the direction of price movement could be. Figure 8.5 and Table 8.2 show the accuracy obtained when the predictions are used to classify the direction of price movement. The Mutli-head CNN-LSTM obtained an accuracy of 46.774%, being outperformed by the MR model by 23.39%. This result is unexpected as the predictions obtained by test 7 achieved the best accuracy thus far with 25.6%. However, closer inspection of the predictions in Figure 8.1 shows that, at several occasions, small price fluctuations of the actual values were not captured.

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	44.355

TABLE 8.2: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7.*

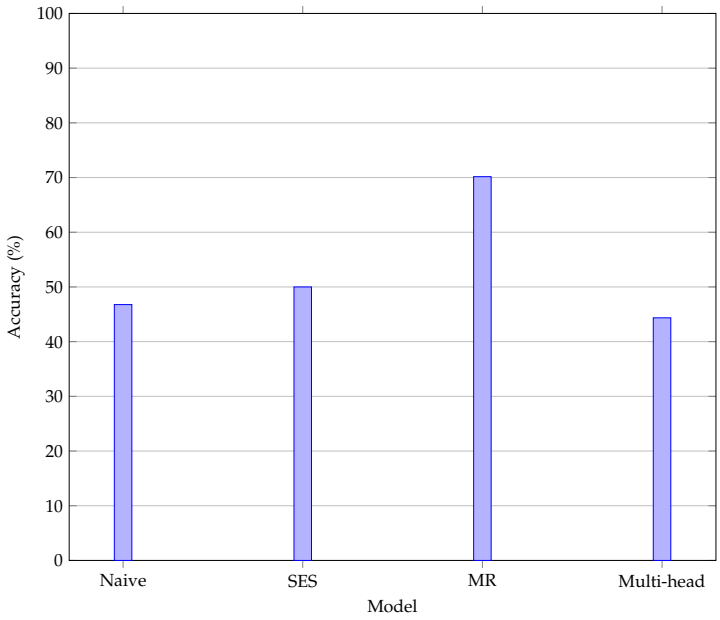


FIGURE 8.5: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7.*

8.2 Results for experiments in Group B

The results that follow were obtained for test 11, the best performing test in Group B. Test 11 had the following properties: 67 CNN neurons per layer, 18 LSTM neurons per layer, 7 CNN layers, 4 LSTM layers, 1 day as input, and inputs which can be seen in Table 3.2 and A.73. Test 11 obtained the highest accuracy, using 1 previous day as input, with an accuracy of 18.4% and MSE of 0.029205. The predictions that were made by test 11 are shown in Figure 8.6 and resemble those made by the LSTM and CNN-LSTM models in Group B. As with previous predictions made, a contributing factor to poor prediction accuracy is the lag element that persists within all the predictions. The MR model outperformed the Multi-head CNN-LSTM model with an accuracy of 30.4%, an increase of 12%, and MSE of 0.004175, a decrease of 0.02503. The MR model outperforming the Multi-head CNN-LSTM model would be expected as the MR model has no lag element.

Looking at the change in accuracy, shown in Figure 8.7(a), can provide a method of observing how predictions are being made. The initial predictions are incorrect with a constant 0% accuracy. The accuracy then increases to 8.33%, after which it fluctuates around an accuracy of 10% and finally increases to an accuracy of 18.4%. The initial period of 0% accuracy could be due to the input features not capturing the relevant information needed for accuracy predictions to be made. The change in error, shown in Figure 8.7(b), supports this possible conclusion as the error fluctuates around -0.2. This shows that the initial predictions, having a similar error, indicate a possible trend in predictions. After the initial predictions, the change in error fluctuates between $|0.4|$ with the exception of two predictions, made on 13-and 14 March 2019, which had errors greater than 0.4.

The influence that network architecture has on prediction accuracy is shown in Figures 8.8(a) and 8.8(b). As network architecture was tested on a subset of those tested in test 1, not all network architectures were tested. Figure 8.8(a) shows the accuracy obtained for different architectures of the CNN and, as expected, several architectures resulted in erratic accuracies.

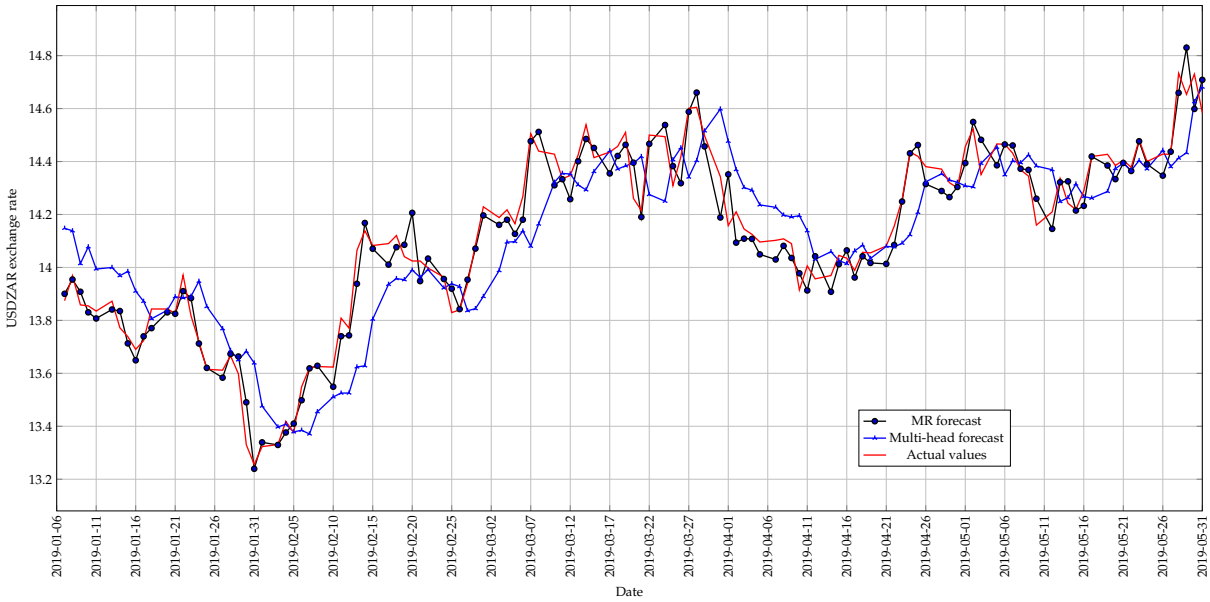
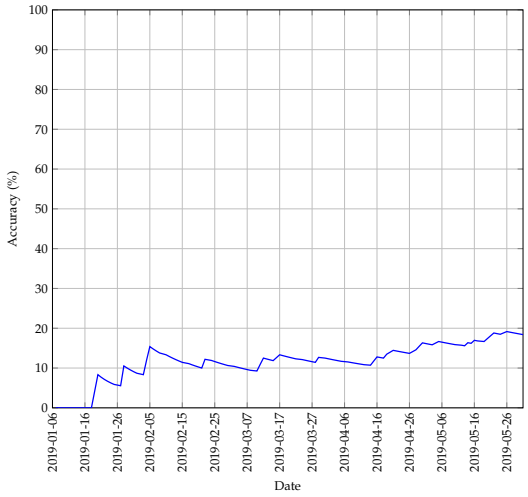
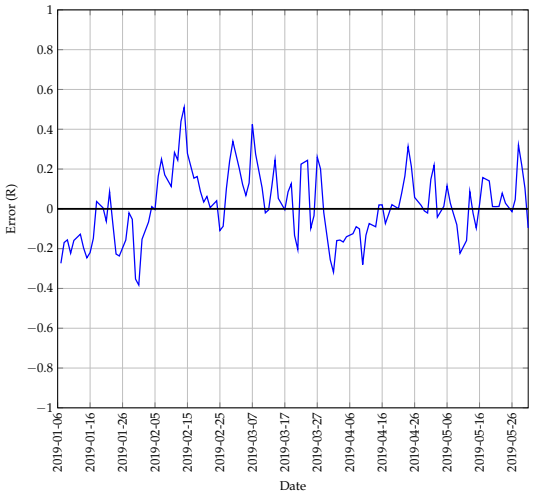


FIGURE 8.6: *The best predictions made by the Multi-head model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set.*



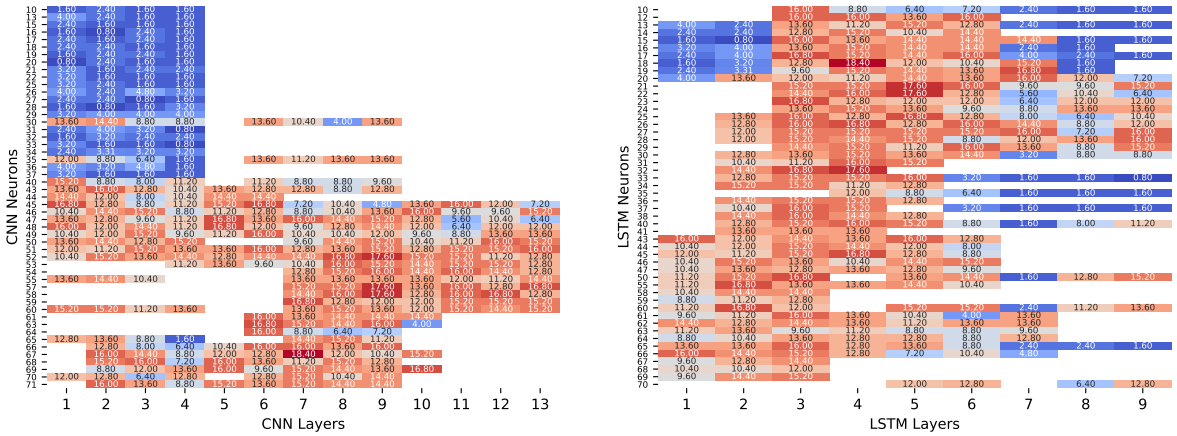
(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE 8.7: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 11.*

This is seen when the number of CNN neurons is less than 40 and the number of CNN layers is less than 5. The accuracy in this specified range is substantially lower than those in surrounding regions. This illustrates the element of unpredictability when tuning the architecture of neural networks. Figure 8.8(b) shows this same unpredictability for different architectures of the LSTM, with accuracies at random architectures obtaining significantly lower accuracies than the surrounding areas. However, although an element of unpredictability exists, it seems as if a possible linear trend exists. This trend could range from architectures with 1 layer and 70 neurons, to 6 layers and 12 neurons. This region obtains the highest accuracies and is encapsulated by lower accuracies. This suggests that an optimal architecture for the LSTM, with further hyper parameter tuning, could exist within this range.



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE 8.8: *A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the Multi-head model.*

Figure 8.9 and Table 8.3 show the change in accuracy when the accuracy threshold is incrementally increased by R0.025. The increase in threshold would suggest that accuracy increases proportionally as the threshold is increased. If the increase in accuracy is transformed to a ratio, the expected increase in accuracy would have the ratios 2, 3, and 4 respectively. However, this is not the case as the observed ratios for the Multi-head CNN-LSTM model were 1.4348, 1.8696, and 2.4348. These observed ratios are substantially lower than those expected and follow the trend of an increasing function with a decreasing rate of change.

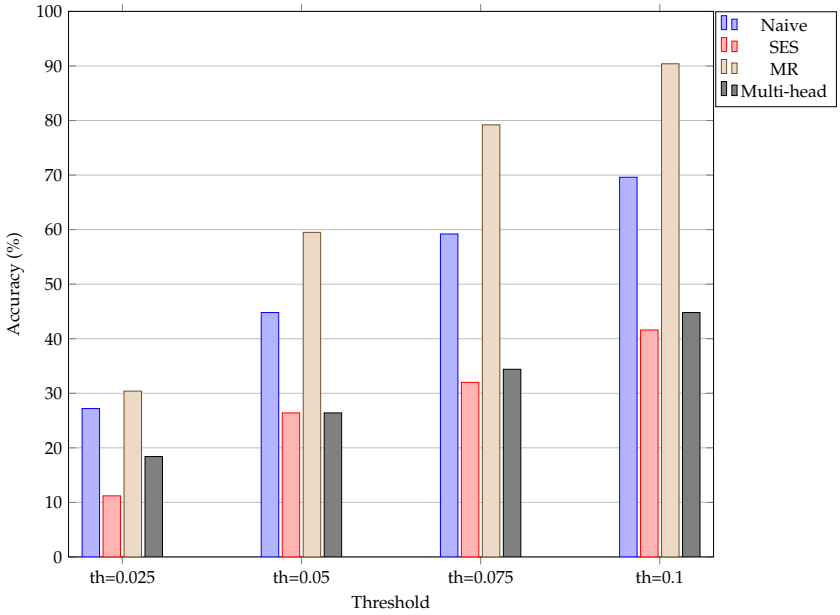


FIGURE 8.9: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	18.4	26.4	34.4	44.8

TABLE 8.3: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11.*

The last factor to consider is the accuracy obtained when the predictions are used in the classification of price movement. Figure 8.10 and Table 8.4 show the accuracy obtained when the predictions are used to classify the direction of price movement. The Multi-head CNN-LSTM obtained an accuracy of 46.774%, being outperformed by the MR model with an accuracy of 70.161%. This is expected as closer inspection of the predictions shows that small fluctuations in the actual values are not captured by the predictions. This results in "smoothed" predictions similar to those that a moving average would make. As a result, when the actual values have small fluctuations in price movement, the predictions overlook this and remain in the same direction.

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	45.968

TABLE 8.4: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11.*

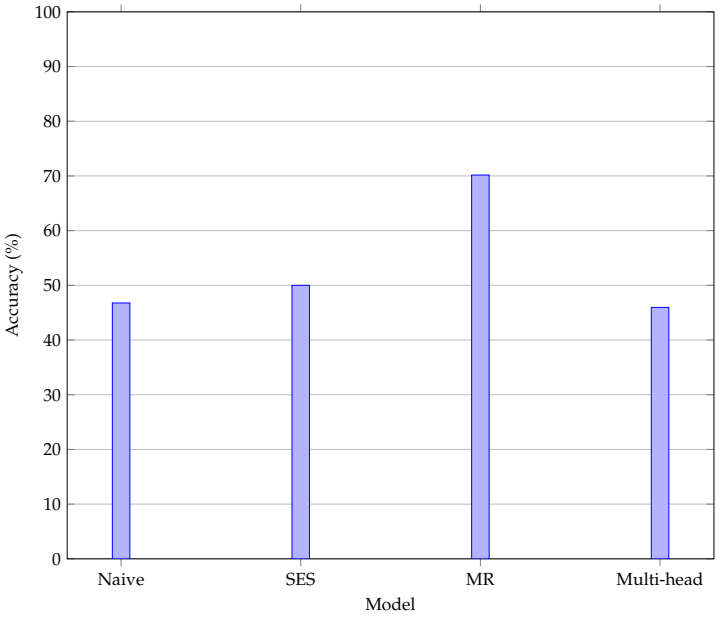


FIGURE 8.10: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11.*

CHAPTER 9

Comparison of results across NN models

The results obtained in Chapters 6, 7, and 8 as well as results from all other tests, can be compared to determine if general trends follow from it. A summary of the best accuracy of the different models for each test is shown in Table 9.1, with squares identifying the best performing neural network for that test. The Multi-head model obtained the best overall accuracy of 25.6% in test 7 when compared to the other NNs. With the addition of Bollinger Bands to the input features of test 1, the multi-head model achieved an increase of 2.4% in accuracy. However, the same cannot be said for the LSTM and CNN-LSTM models, which saw a decrease of 1.6% in accuracy.

When the MSE of the models is considered in Table 9.2, the same pattern does not exist. The MSE for the Multi-head model and LSTM model for test 7 increased by 0.0021 and 0.00037 respectively, whereas the CNN-LSTM model saw a decrease of 0.00133. A possible explanation for this could be due to the input features not capturing all the relevant information during specific time periods. In this scenario, it could lead to the models making an incorrect prediction that has a large error in specific time periods, thus increasing the overall MSE of the model, but the predictions in the remaining time periods are correct, thus increasing the prediction accuracy.

Model	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**
Naive	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2	27.2
SES	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6	5.6
MR	30.4	30.4	30.4	30.4	30.4	30.4	30.4	30.4	30.4	30.4	30.4	30.4
LSTM	22.4	20	20	17.6	17.6	16	20.8	19.2	16.8	20.8	17.6	8.8
CNN-LSTM	24.8	22.4	21.6	22.4	22.4	17.6	23.2	21.6	18.4	22.4	18.4	4.8
Multi-head	23.2	22.4	20	20.8	16	17.6	25.6	20	20	20.8	18.4	9.6

TABLE 9.1: *The accuracy, in percentage, obtained by all the models for all the tests conducted.*

Although the Multi-head model achieved the highest prediction accuracy overall, the CNN-LSTM model obtained the highest prediction accuracy for 7 of the 12 tests conducted. This

illustrates that different NN architectures are more sensitive to different input features. However, the CNN-LSTM obtained its highest prediction accuracy of 24.8% for test 1. This is unexpected as test 1 had the most basic features, indicating that the addition of input features in this case provides no performance benefit. Test 7 performed second best, with an accuracy of 23.2%. This could indicate that expanding parameter tuning for the CNN-LSTM could possibly result in a network that outperforms the 25.6% accuracy obtained by the Multi-head model. The same could be said for the LSTM model which had tests 7 and 10 perform second best after test 1.

Model	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**
Naive	0.01139	0.01139	0.01139	0.01139	0.01139	0.01139
SES	0.11798	0.11798	0.11798	0.11798	0.11798	0.11798
MR	0.00417	0.00417	0.00417	0.00417	0.00417	0.00417
LSTM	0.02263	0.02550	0.02200	0.02291	0.03606	0.03616
CNN-LSTM	0.02447	0.02781	0.02054	0.02144	0.02940	0.03210
Multi-head	0.02272	0.02549	0.02165	0.02205	0.02464	0.03246

Model	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**
Naive	0.01139	0.01139	0.01139	0.01139	0.01139	0.01139
SES	0.11798	0.11798	0.11798	0.11798	0.11798	0.11798
MR	0.00417	0.00417	0.00417	0.00417	0.00417	0.00417
LSTM	0.02300	0.02395	0.02555	0.02294	0.03643	0.21353
CNN-LSTM	0.02314	0.02252	0.03015	0.02203	0.03853	0.28048
Multi-head	0.02482	0.02193	0.02151	0.02137	0.02921	0.18336

TABLE 9.2: *The MSE obtained by all the models for all the tests conducted.*

Although changing the accuracy threshold was not the primary measure of performance, it provides another method to analyse the effect that input features have. Table 9.3 and 9.4 show the change in accuracy obtained for the four best tests when the accuracy threshold is increased. From Table 9.3, test 1 and 5 have similar performance patterns. For test 1, the CNN-LSTM performs the best for the first two thresholds, the Multi-head performs best for the third threshold, and the LSTM model performs best for the final threshold. For test 5, the only difference is that the Multi-head model performs best for the last two thresholds. The CNN-LSTM outperforming the other models for the first two thresholds indicates that the CNN-LSTM has, on average, predictions which are closer to the actual values. The CNN-LSTM therefore benefits the most from using the basic input features. The LSTM and Multi-head models therefore do not benefit as much from the basic inputs as their accuracies are only notable once the threshold has more than doubled. For test 5, the overall accuracy for all models across all thresholds decreases when compared to test 1. Including the Heiken-Ashi indicator in the input features does therefore not benefit any of the models. However, for the first two thresholds the CNN-LSTM model outperforms the other NNs, indicating that the CNN-LSTM benefits the most from the addition of the Heiken-Ashi indicator.

The dominant NN, from Table 9.3, changes to the Multi-head model in Table 9.4. For test 7, the Multi-head model outperforms all NNs for each threshold, having a tie with the CNN-LSTM for the third threshold. The addition of Bollinger Bands for test 7 therefore benefits the Multi-head model substantially when compared to the Multi-head model results in Table 9.3. However, the addition of Bollinger bands negatively impacts the accuracy of the LSTM for all thresholds while the CNN-LSTM obtains a decrease for the first two thresholds and an increase in accuracy for the third threshold. For test 11, all features were given to all the NNs. From Table 9.4,

Model	Test 1 threshold (%)				Test 5 threshold (%)			
	0.025	0.05	0.075	0.1	0.025	0.05	0.075	0.1
Naive	27.2	44.8	59.2	69.6	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4	30.4	59.5	79.2	90.4
LSTM	22.4	32	48	56	17.6	24.8	34.4	44.8
CNN-LSTM	24.8	32.8	39.2	52	22.4	32	24.4	48.8
Multi-head	23.2	31.2	49.6	54.4	16	24.8	40.8	50.4

TABLE 9.3: *The accuracy obtained by the different models for test 1 and test 5 with varying accuracy thresholds.*

the accuracy for all NNs and thresholds decreased. However, the CNN-LSTM returns as the dominant NN, obtaining the highest accuracy for the first three thresholds but losing to the Multi-head model which outperformed with the last threshold.

Model	Test 7 threshold (%)				Test 11 threshold (%)			
	0.025	0.05	0.075	0.1	0.025	0.05	0.075	0.1
Naive	27.2	44.8	59.2	69.6	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4	30.4	59.5	79.2	90.4
LSTM	19.2	31.2	42.4	53.6	8.8	11.2	13.6	15.2
CNN-LSTM	23.2	33.6	44.0	52.0	18.4	28.0	35.2	44.0
Multi-head	25.6	36.0	44.0	55.2	18.4	26.4	34.4	44.8

TABLE 9.4: *The accuracy obtained by the different models for test 7 and test 11 with varying accuracy thresholds.*

As with the change in accuracy thresholds, classifying the direction of price movement was not considered a primary performance measure, but rather another metric to compare the effect that features have. Table 9.5 shows the accuracy for all the models for all tests when classifying the direction of price movement. As with the previous metrics, the CNN-LSTM continues to outperform all other NNs, obtaining the highest accuracy for 8 of the 12 tests. The largest increase in accuracy for the CNN-LSTM of 14.516% was obtained for test 12, with the addition of PCA components. The LSTM model obtained its highest increase in accuracy of 11.29% for test 10, with the addition of first and second differences and the Multi-head model, like the CNN-LSTM, also saw its highest increase of 7.258% for test 12. Considering all the accuracies obtained for all tests of all models, reveals that the LSTM saw no decline in accuracy, the CNN-LSTM saw one decline in accuracy for test 6 and the Multi-head model saw a decline in accuracy for tests 2, 5, 8 and 10. This suggests that the Multi-head model has an increased sensitivity towards input features when compared to the other NNS.

Despite an existing sensitivity to network architecture, these results show that NNs are also sensitive to the input features. Despite the large amount of parameter tuning that can still occur, several tests see constant improvements throughout all the models. The overall top performing tests for each NN are shown in Table 9.6, and reveal that tests 1, 7, 10 and 12 generate the best results.

Model	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**
Naive	46.774	46.774	46.774	46.774	46.774	46.774
SES	50.000	50.000	50.000	50.000	50.000	50.000
MR	70.161	70.161	70.161	70.161	70.161	70.161
LSTM	42.742	46.774	43.548	49.194	44.355	45.161
CNN-LSTM	45.968	49.194	46.774	46.774	49.194	45.161
Multi-head	44.355	41.935	45.968	45.161	42.742	45.968

Model	**Test 7**	**Test 8**	**Test 9**	**Test 10**	**Test 11**	**Test 12**
Naive	46.774	46.774	46.774	46.774	46.774	46.774
SES	50.000	50.000	50.000	50.000	50.000	50.000
MR	70.161	70.161	70.161	70.161	70.161	70.161
LSTM	46.774	49.194	45.968	54.032	50.000	46.774
CNN-LSTM	50.806	49.194	53.226	46.774	49.194	60.484
Multi-head	44.355	42.742	50.000	43.548	45.968	51.613

TABLE 9.5: *The accuracy obtained by all the models for all the tests conducted when the direction of price movement is considered.*

Metric	LSTM	CNN-LSTM	Multi-head
Highest prediction accuracy	Test 1	Test 1	Test 7
Second highest prediction accuracy	Tests 7 & 10	Test 7	Test 1
Highest classification accuracy	Test 10	Test 12	Test 12

TABLE 9.6: *The overall best performing tests for each NN.*

CHAPTER 10

Conclusion

Contents

10.1 Recommendations 81
10.2 Objectives 82
10.3 Future work 82
10.3.1 Deepening the study *83*
10.3.2 Broadening the study *83*

The effect and importance of input features was investigated on three different neural network architectures. Using the prediction accuracy, MSE, adjusting the accuracy threshold and classification accuracy, a comparison was done between the different tests, which used different input features, and the overall best performing NNs. As NNs are sensitive to network architectures, several architectures were also investigated for each input feature, thus allowing the opportunity for each test to find a good configuration.

10.1 Recommendations

The numerous possibilities and permutations of network architectures, hyper parameters and input features renders it impossible to determine the overall best input feature. However, several possible conclusions can be drawn from the results obtained in Chapters 6, 7 and 8 as well as the comparison in Chapter 9.

It was seen that test 1, using basic input features, resulted in the highest prediction accuracy for the LSTM and CNN-LSTM models, and second-highest prediction accuracy for the Multi-head model. This would suggest that the open, high, low and close are important features and affect the performance of the networks. Test 7, which had basic input features and Bollinger Bands, resulted in the highest overall prediction accuracy obtained by the Multi-head model, and the second-highest prediction accuracy for the LSTM and CNN-LSTM models. By combining the elements of Bollinger Bands to the basic input features, it would suggest that Bollinger bands have the greatest impact on prediction accuracy, and are therefore important to include when modelling. However, the LSTM model could benefit from the addition of Bollinger Bands and the first and second differences as both feature sets resulted in the second-highest prediction accuracy. Therefore, Bollinger Bands and first and second differences are considered important for the LSTM model.

When classifying the direction of price movement, test 10 and 12 resulted in the highest classification accuracy, using first and second differences and PCA components respectively. In the case of the CNN-LSTM and Multi-head model, the basic features with the addition of PCA components resulted in the best classification accuracies. Therefore, adding PCA components when classifying price movement is considered important for the CNN-LSTM and Multi-head model. However, when the LSTM is considered, the basic features with the addition of first and second differences resulted in the best classification accuracy and can therefore be considered important.

10.2 Objectives

In Chapter 1 the following objectives were identified:

1. Provide a background to finance and the importance of knowing what the exchange rate might potentially do;
2. Provide a background to machine learning and the applications in finance;
3. Describe the data and models that were used and implemented;
4. Describe the process of how experimentation was done;
5. Investigate the forecasting performance of standard statistical methods;
6. Compare results between the different NNs as well as how the input features affected those results;
7. Discuss potential directions that could be taken for further research.

Objective 1 is achieved in Chapter 1, where a detailed background to finance and the importance of being able to predict the movement of the exchange rate was given. Objective 2 was also achieved in Chapter 1 where a background of machine learning and its applications in finance was described. In Chapter 3 the data and models that were used and implemented were outlined and described, thus achieving Objective 3. In Chapter 3, Objective 4 was achieved where the chosen methodology for experimentation was provided. Objective 5 was achieved in Chapter 5 where the forecasting performance of standard statistical methods was investigated. In Chapters 6, 7, and 8, the results between the different NN models were given; and in Chapter 9 a comparison of the effect of different input features was given; thus achieving Objective 6. Finally, in Chapter 10 possible directions for future research was given, achieving Objective 7.

10.3 Future work

The scope of this study was focused on testing different input features and measuring their effect on performance. Two factors can be considered for future work: deepening the study by considering the effect of several hyper parameters, thus expanding the scope, and broadening the study by considering different methodologies and data manipulation techniques.

10.3.1 Deepening the study

A factor to deepen this study would be to investigate the effect of hyper parameters on prediction accuracy. Although input features are important for neural networks, their effect on prediction accuracy will be further highlighted as hyper parameters are changed. This could lead to possible conclusions such as "different activation functions have no effect on prediction accuracy when Bollinger Bands are added to basic input features". Thus leading to an improved and narrower selection of input features, as well as improved NN architectures.

10.3.2 Broadening the study

A natural continuation of this study would be to consider decomposing the time series data of the input features, removing noise from the data, and then testing the input features again. This would increase the predictability of the daily closing price of the USD/ZAR exchange rate and lead to an improvement in prediction accuracy.

Another approach to broaden the study would be to investigate pattern detection within the daily closing price. This could be done by training a CNN to detect specific patterns within the data. If a specific pattern is recognised, it could indicate either an increase or decrease in price movement. However, training a CNN for this purpose would require either supervised or semi-supervised learning and a labelled dataset which would have to be generated. The output of the CNN, the detection of a specific pattern, could then be given as an input to a LSTM which could then predict the closing price.

APPENDIX A

Appendix

A.1 LSTM model results

This section contains the results obtained by the LSTM model.

A.1.1 Test 1 results

The results obtained for test 1 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.1 shows the predictions made by the LSTM model,
- Figure A.2(a) and Figure A.2(b) show the change in accuracy and error as predictions are made,
- Figure A.3 shows the accuracy obtained for different neural network architectures,
- Figure A.4 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.5 shows the accuracy obtained when predicting the direction of price movement,
- Table A.1 represents the results obtained when the accuracy threshold is changed,
- Table A.2 illustrates the accuracy obtained when the direction of price movement is predicted.

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	22.4	32.0	48.0	56.0

TABLE A.1: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1.*

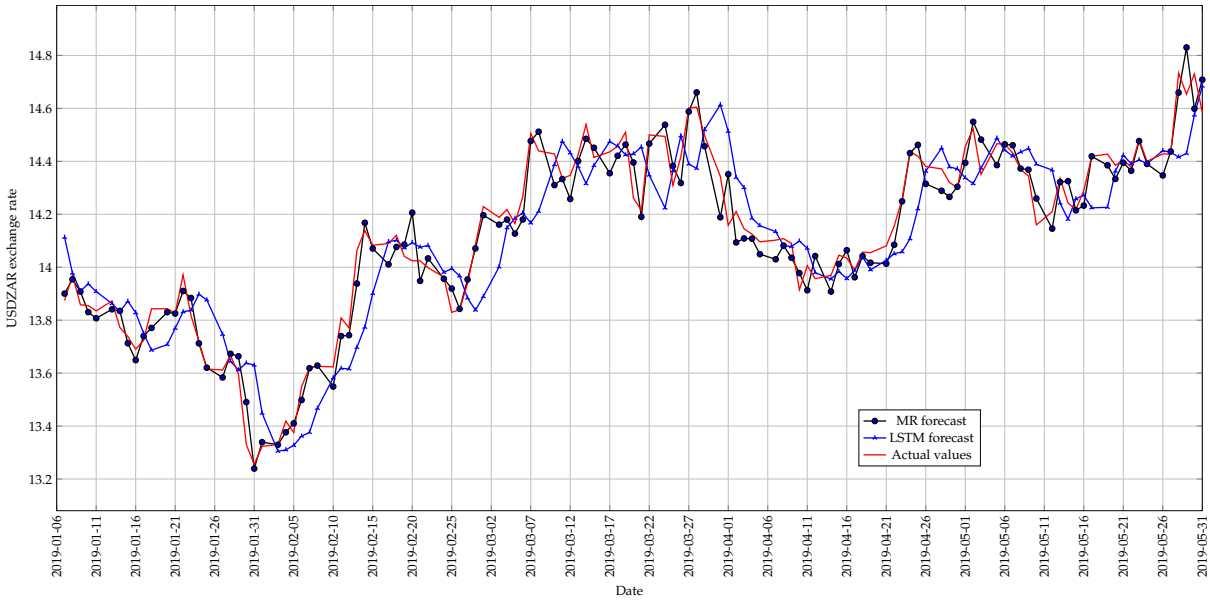
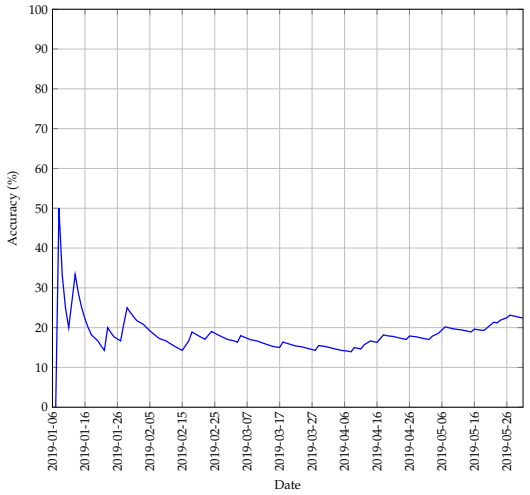
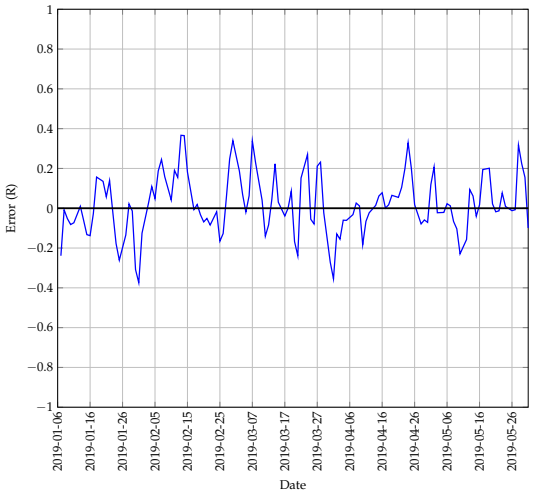


FIGURE A.1: *The best predictions made by the LSTM model for test 1 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.2: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 1.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	45.968

TABLE A.2: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1.*

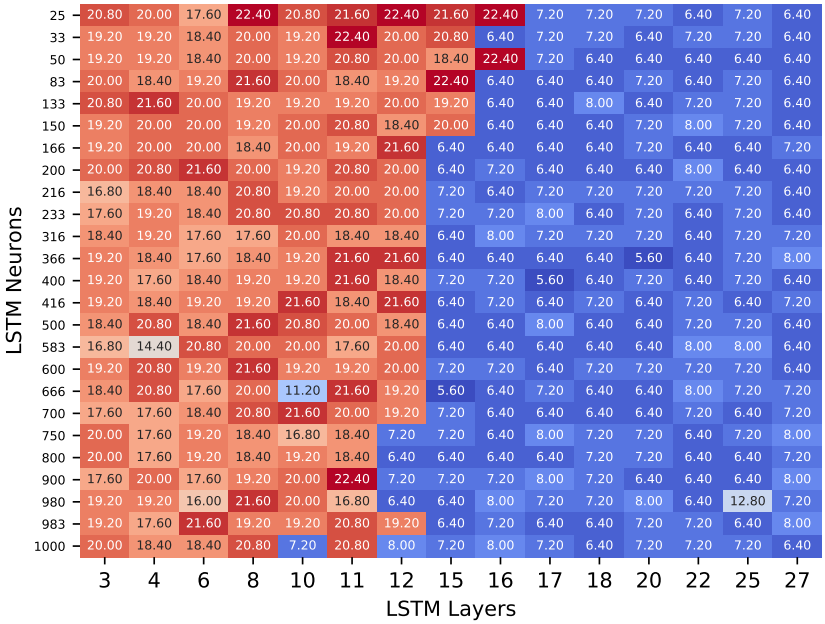


FIGURE A.3: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 1 with 1 previous day as input.*

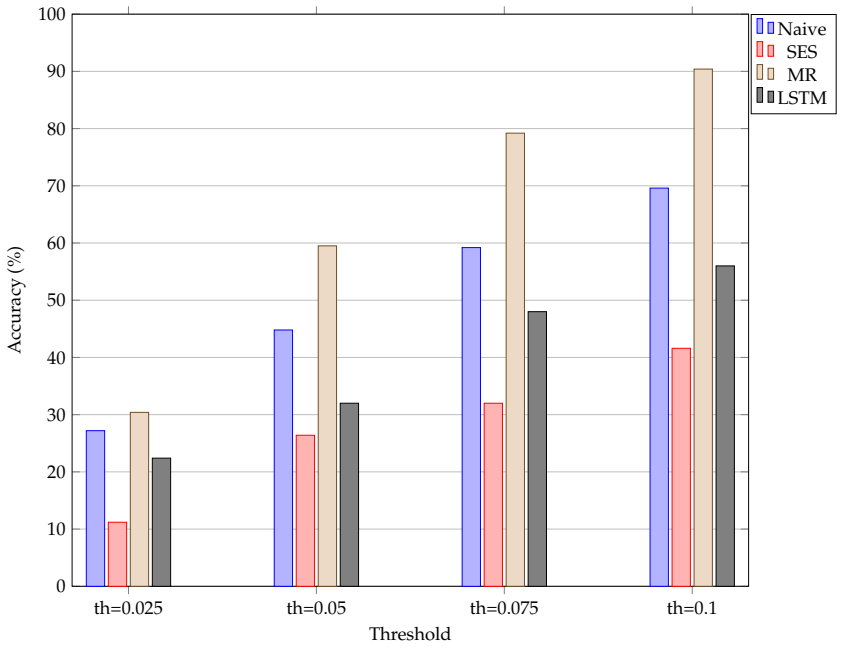


FIGURE A.4: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025.*

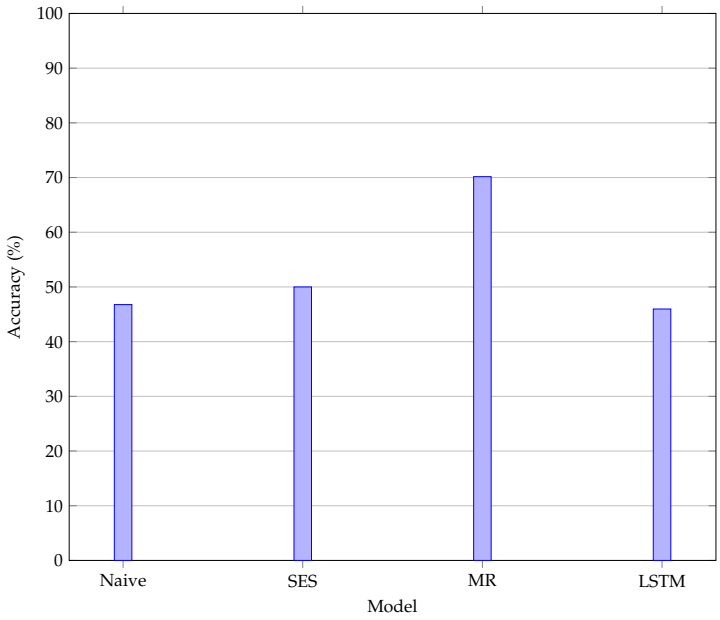


FIGURE A.5: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1.*

A.1.2 Test 2 results

The results obtained for test 2 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.6 shows the predictions made by the LSTM model,
- Figure A.7(a) and Figure A.7(b) show the change in accuracy and error as predictions are made,
- Figure A.8 shows the accuracy obtained for different neural network architectures,
- Figure A.9 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.10 shows the accuracy obtained when predicting the direction of price movement,
- Table A.3 represents the results obtained when the accuracy threshold is changed,
- Table A.4 illustrates the accuracy obtained when the direction of price movement is predicted.

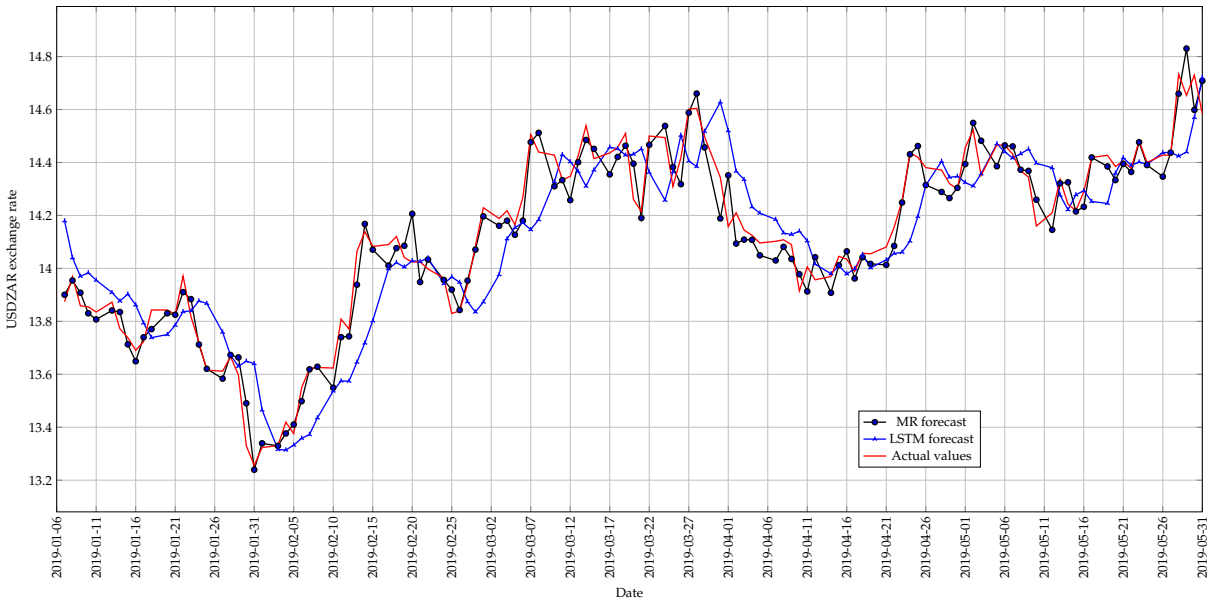
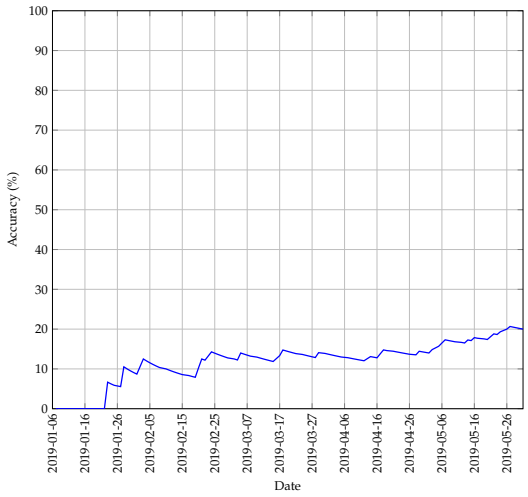
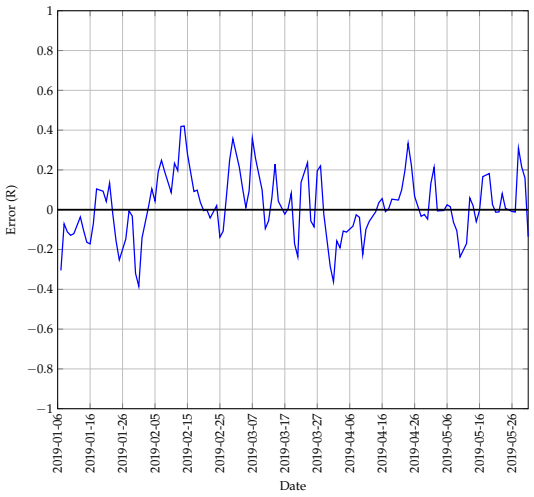


FIGURE A.6: *The best predictions made by the LSTM model for test 2 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.7: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 2.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	20.0	31.2	41.6	52.0

TABLE A.3: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2.*

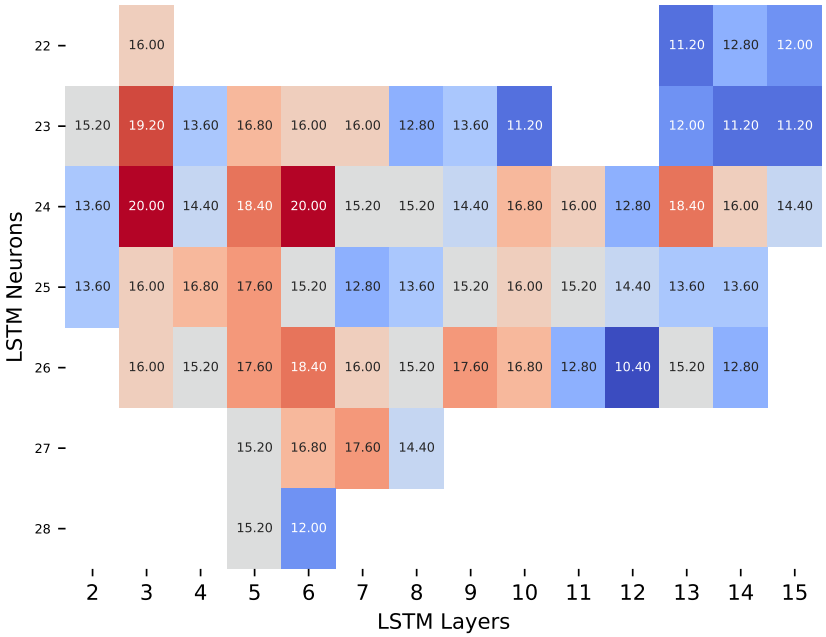


FIGURE A.8: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 2 with 1 previous day as input.*

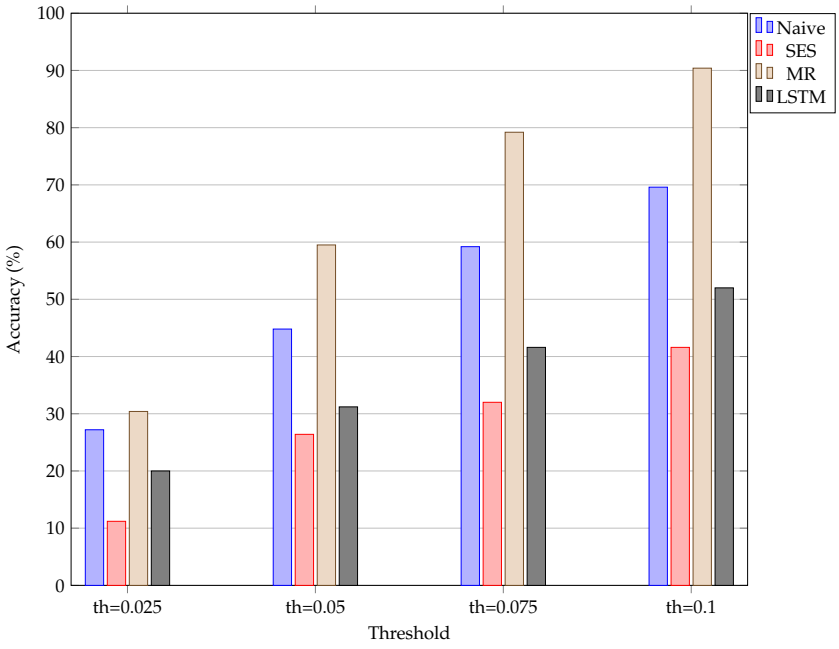


FIGURE A.9: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025.*

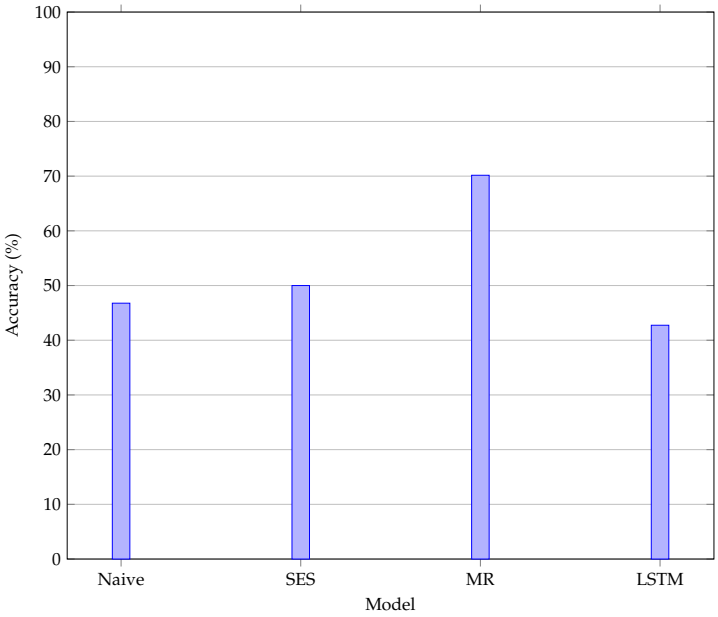


FIGURE A.10: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
LSTM	42.742

TABLE A.4: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2.*

A.1.3 Test 3 results

The results obtained for test 3 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.11 shows the predictions made by the LSTM model,
- Figure A.12(a) and Figure A.12(b) show the change in accuracy and error as predictions are made,
- Figure A.13 shows the accuracy obtained for different neural network architectures,
- Figure A.14 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.15 shows the accuracy obtained when predicting the direction of price movement,
- Table A.5 represents the results obtained when the accuracy threshold is changed,
- Table A.6 illustrates the accuracy obtained when the direction of price movement is predicted.

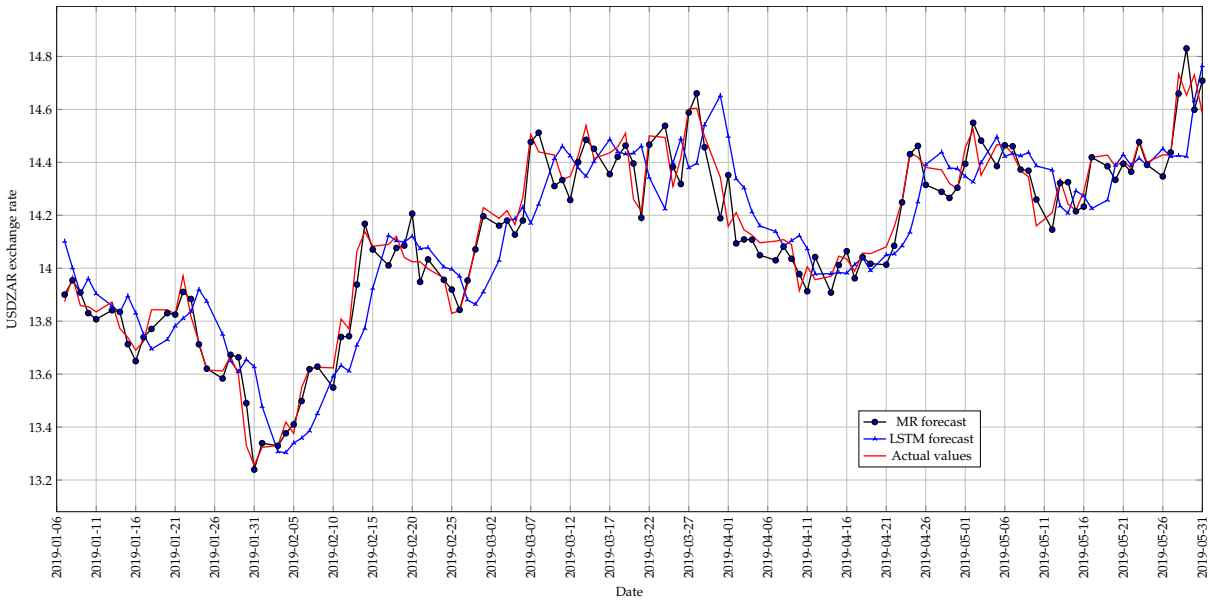
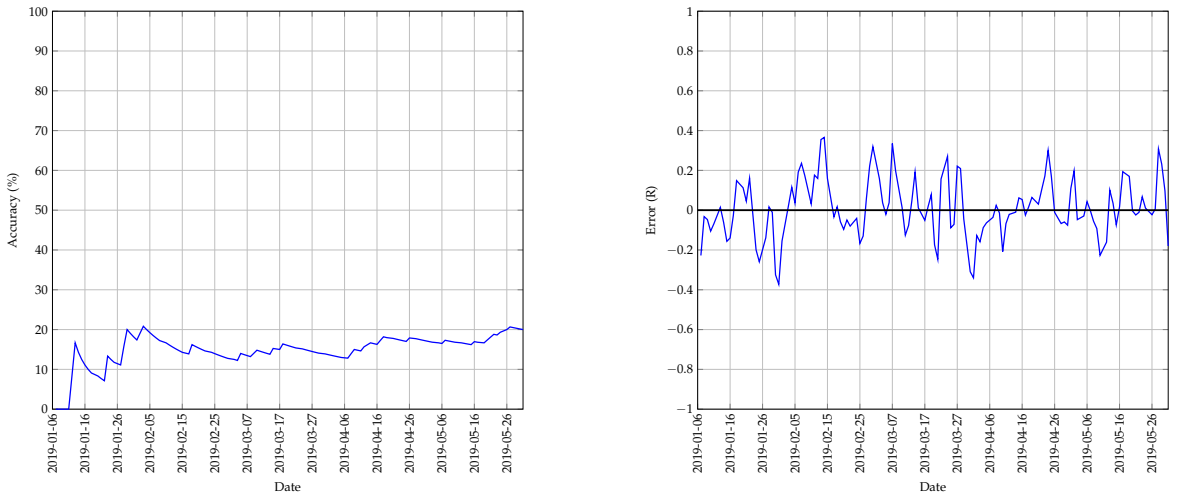


FIGURE A.11: *The best predictions made by the LSTM model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	20.0	35.2	48.0	55.2

TABLE A.5: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.12: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 3.*

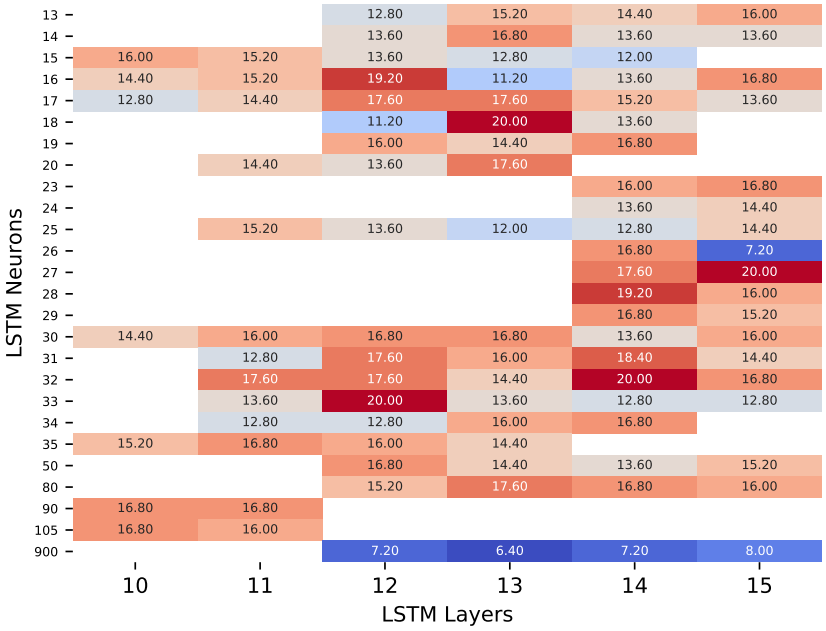


FIGURE A.13: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 3 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	46.774

TABLE A.6: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3.*

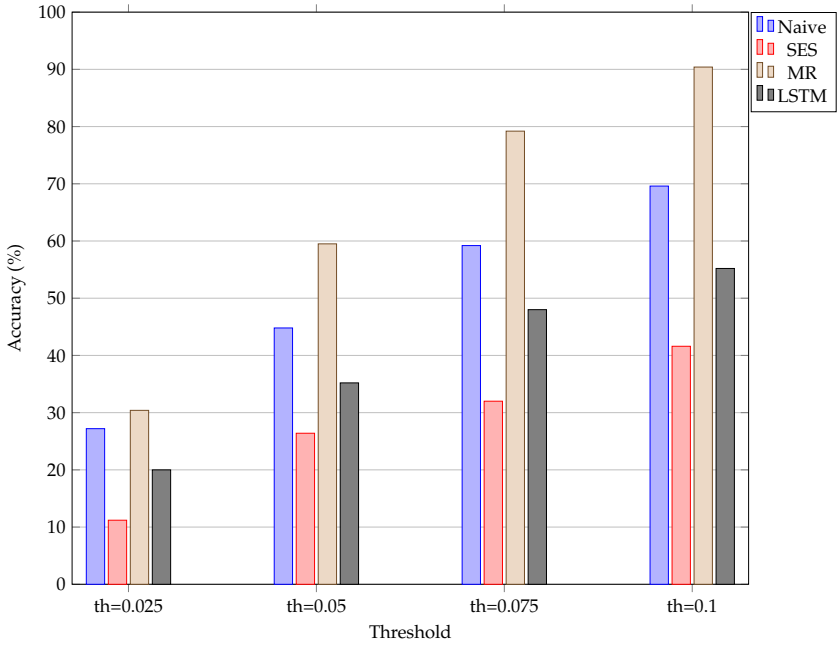


FIGURE A.14: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025.*

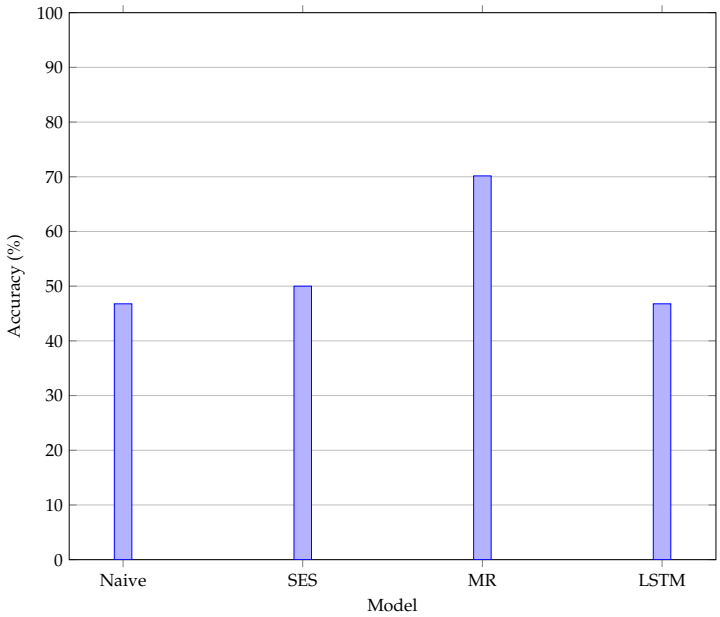


FIGURE A.15: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3.*

A.1.4 Test 4 results

The results obtained for test 4 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.16 shows the predictions made by the LSTM model,
- Figure A.17(a) and Figure A.17(b) show the change in accuracy and error as predictions are made,
- Figure A.18 shows the accuracy obtained for different neural network architectures,
- Figure A.19 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.20 shows the accuracy obtained when predicting the direction of price movement,
- Table A.7 represents the results obtained when the accuracy threshold is changed,
- Table A.8 illustrates the accuracy obtained when the direction of price movement is predicted.

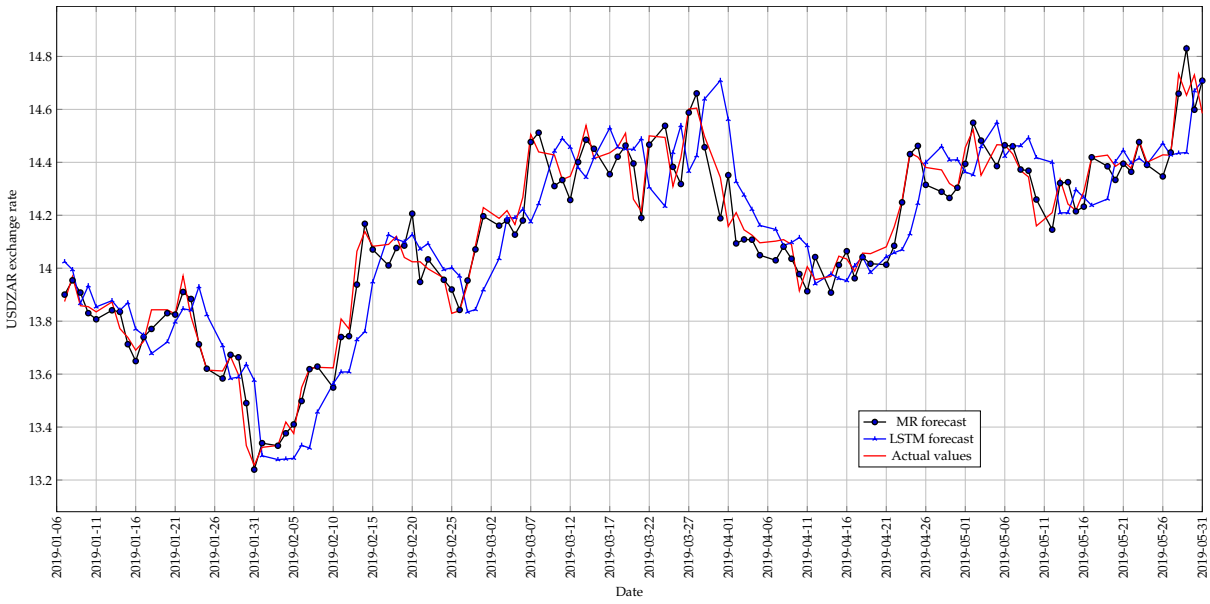
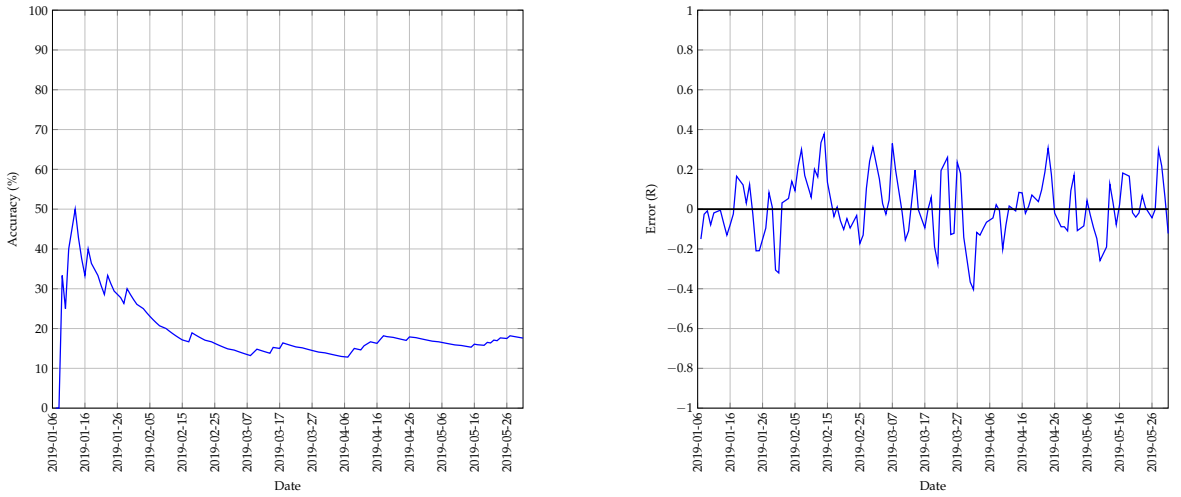


FIGURE A.16: *The best predictions made by the LSTM model for test 4 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	17.6	31.2	38.4	52.8

TABLE A.7: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.17: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 4.*

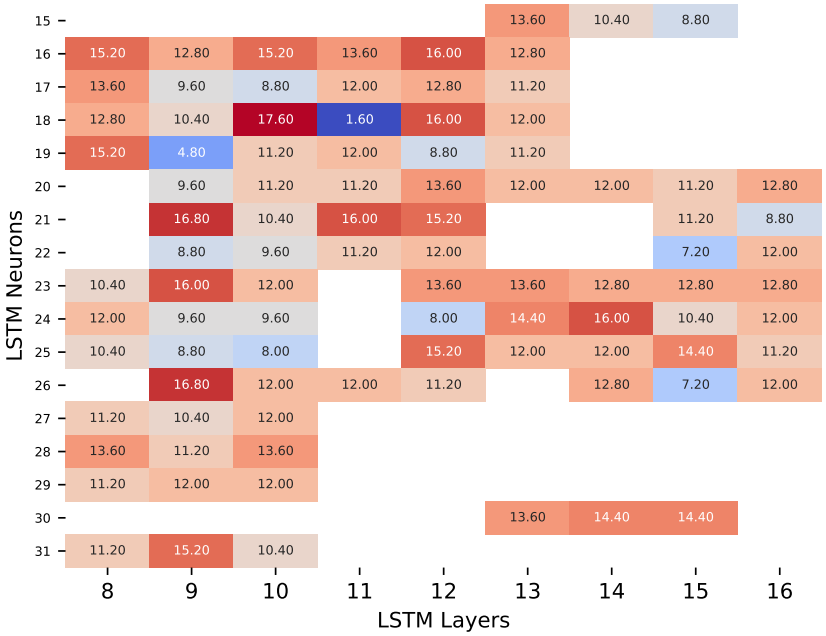


FIGURE A.18: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 4 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	43.548

TABLE A.8: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4.*

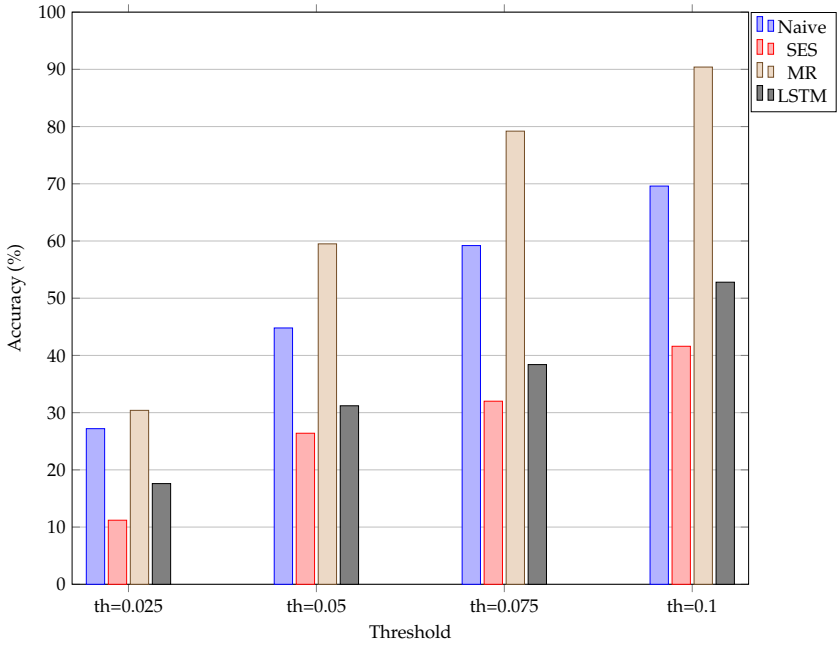


FIGURE A.19: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025.*

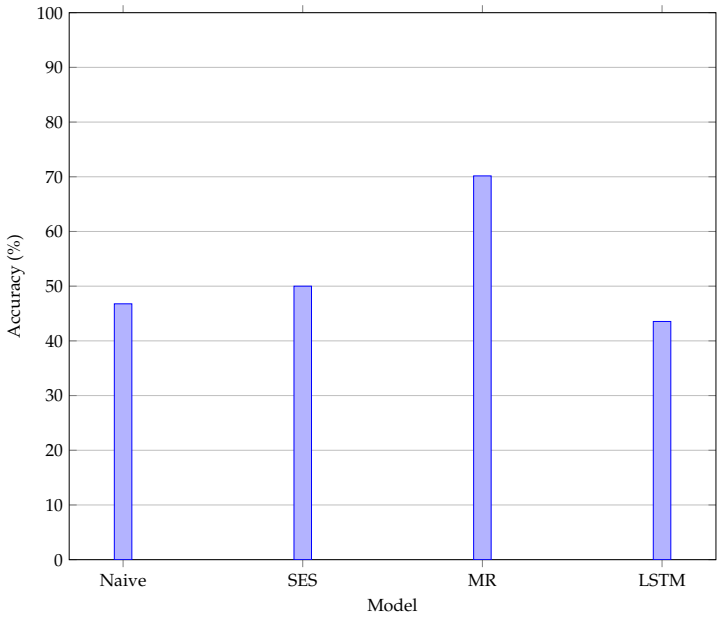


FIGURE A.20: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4.*

A.1.5 Test 5 results

The results obtained for test 5 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.21 shows the predictions made by the LSTM model,
- Figure A.22(a) and Figure A.22(b) show the change in accuracy and error as predictions are made,
- Figure A.23 shows the accuracy obtained for different neural network architectures,
- Figure A.24 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.25 shows the accuracy obtained when predicting the direction of price movement,
- Table A.9 represents the results obtained when the accuracy threshold is changed,
- Table A.10 illustrates the accuracy obtained when the direction of price movement is predicted.

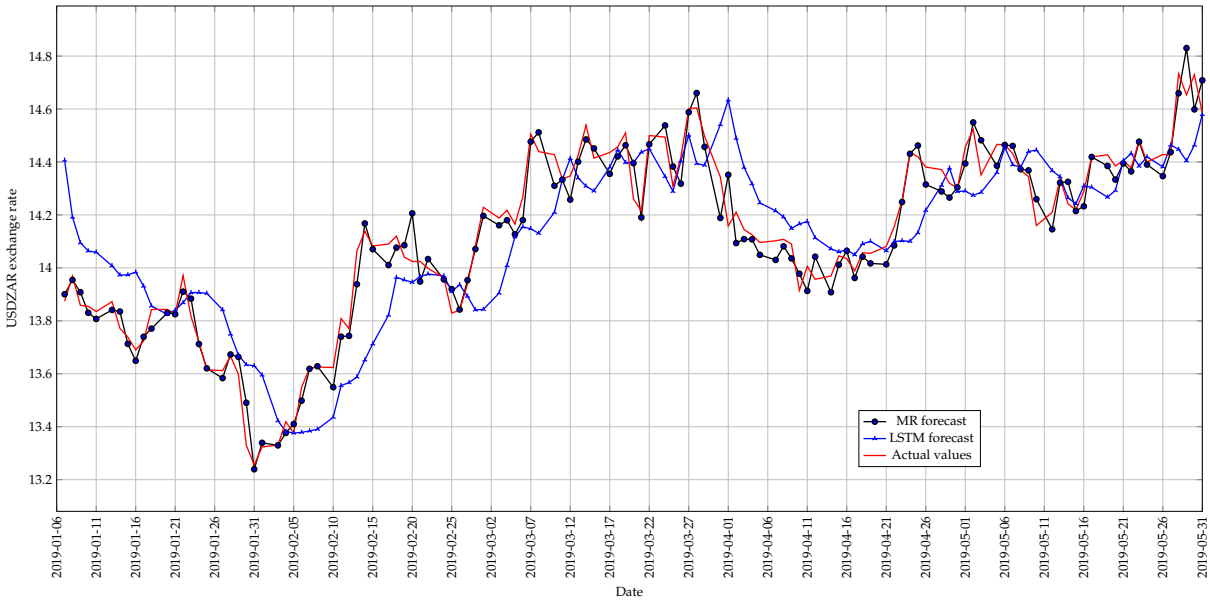
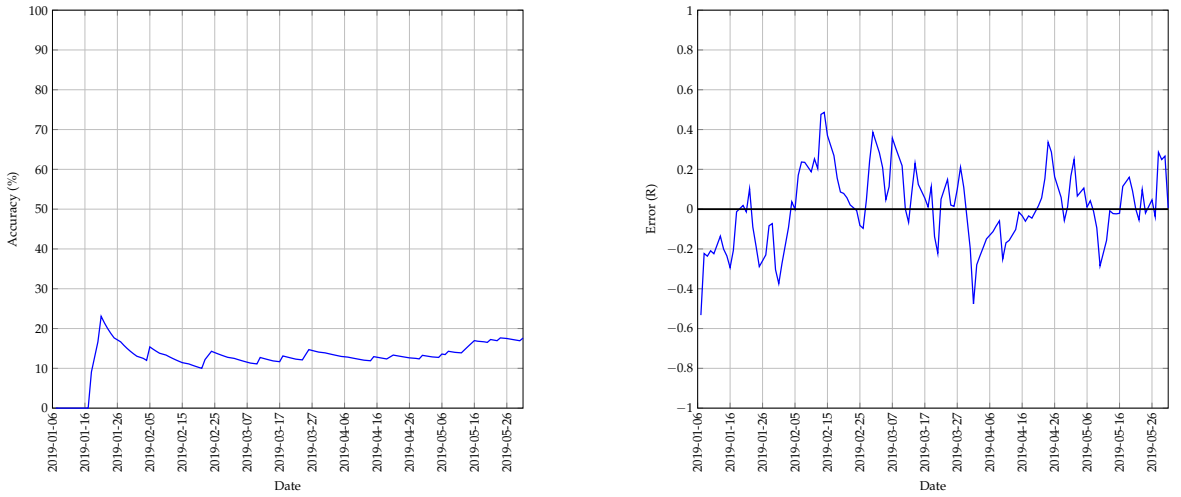


FIGURE A.21: *The best predictions made by the LSTM model for test 5 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	17.6	24.8	34.4	44.8

TABLE A.9: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.22: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 5.*

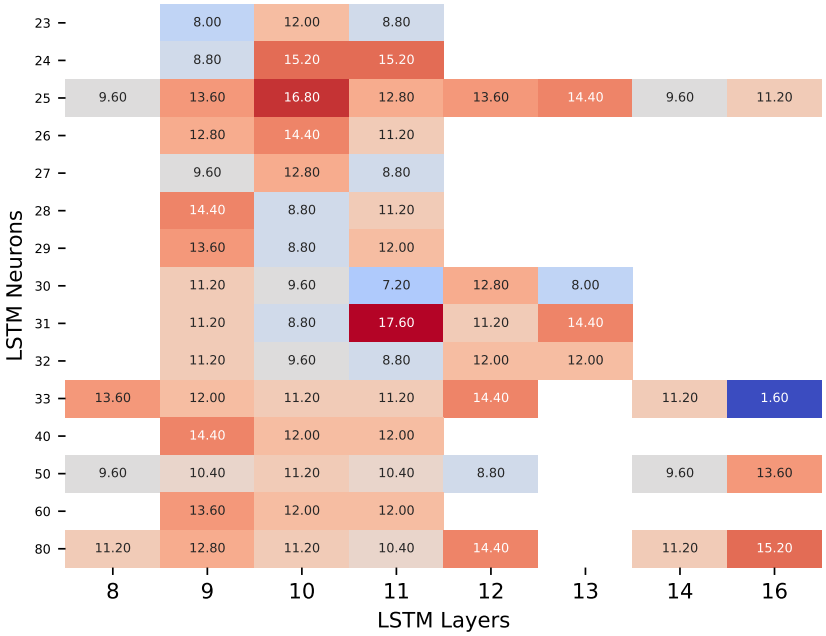


FIGURE A.23: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 5 with 2 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	49.194

TABLE A.10: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5.*

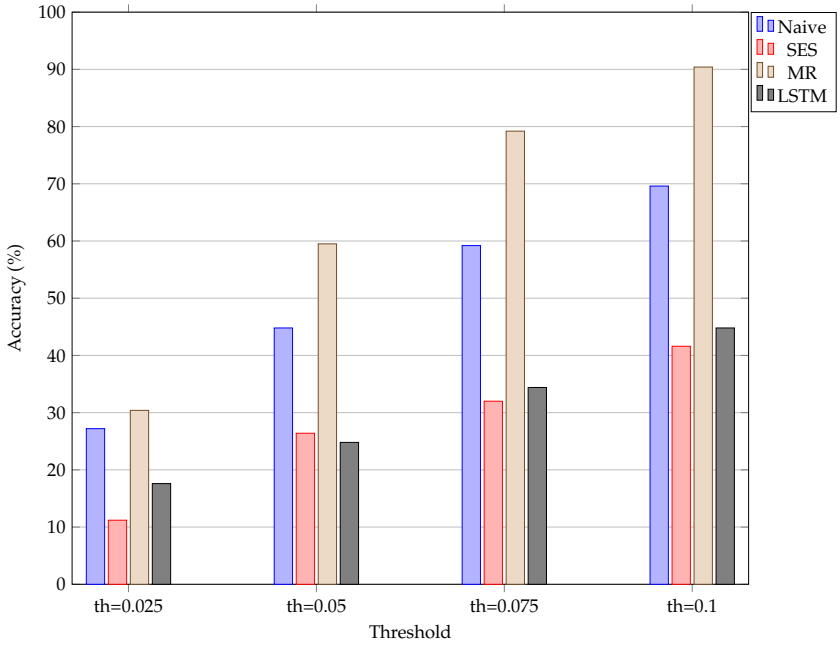


FIGURE A.24: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025.*

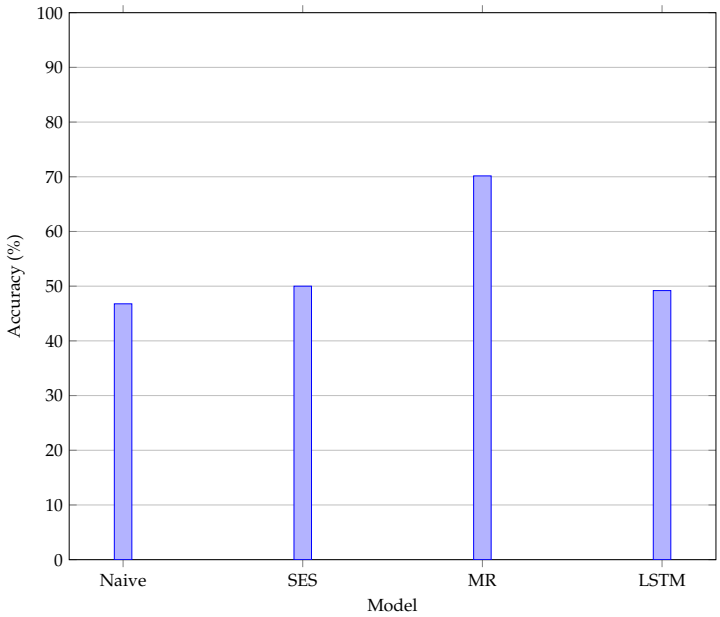


FIGURE A.25: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5.*

A.1.6 Test 6 results

The results obtained for test 6 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.26 shows the predictions made by the LSTM model,
- Figure A.27(a) and Figure A.27(b) show the change in accuracy and error as predictions are made,
- Figure A.28 shows the accuracy obtained for different neural network architectures,
- Figure A.29 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.30 shows the accuracy obtained when predicting the direction of price movement,
- Table A.11 represents the results obtained when the accuracy threshold is changed,
- Table A.12 illustrates the accuracy obtained when the direction of price movement is predicted.

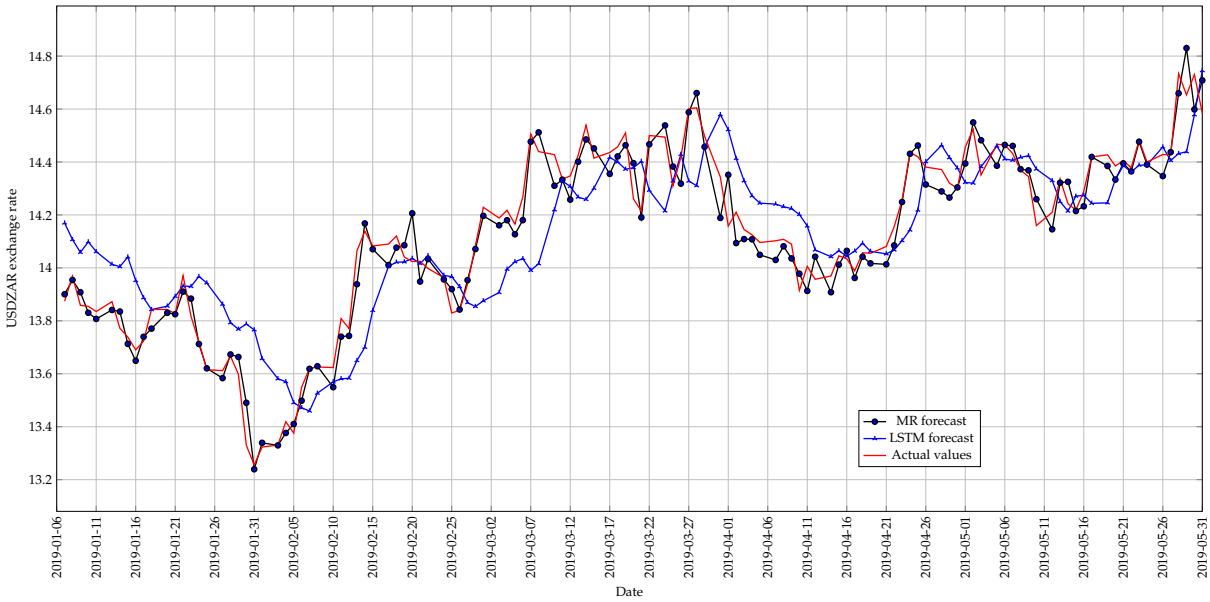
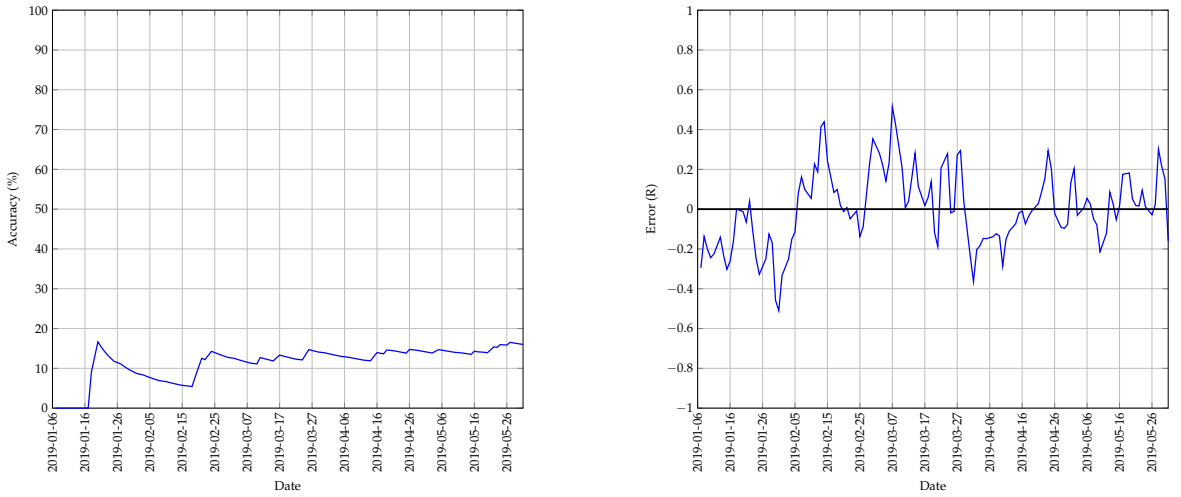


FIGURE A.26: *The best predictions made by the LSTM model for test 6 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	16.0	24.8	32.0	41.6

TABLE A.11: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.27: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 6.*

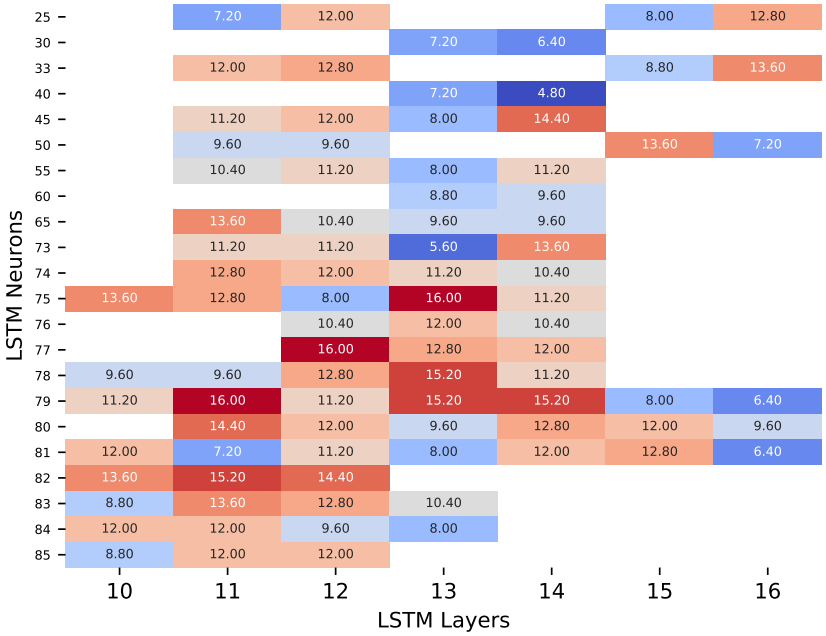


FIGURE A.28: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 6 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	44.355

TABLE A.12: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6.*

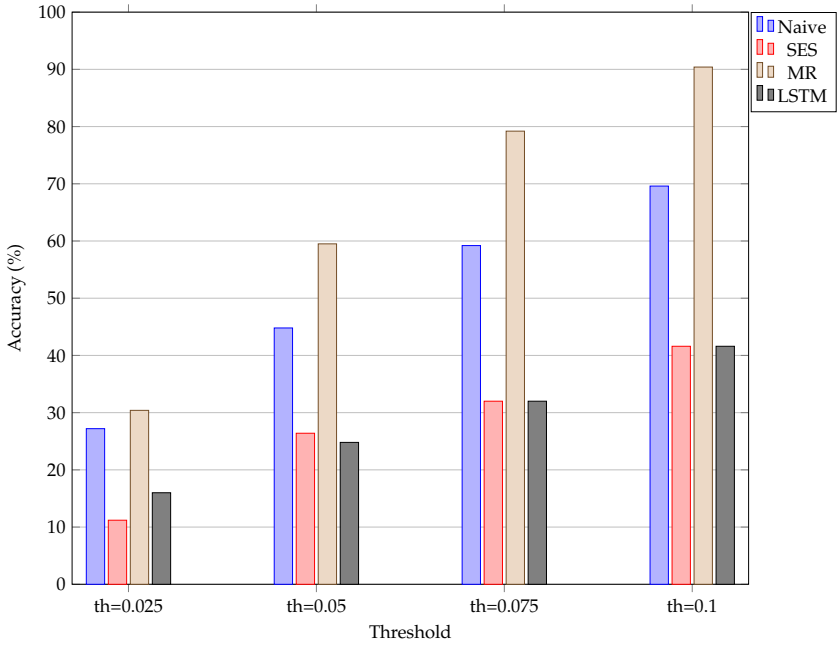


FIGURE A.29: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025.*

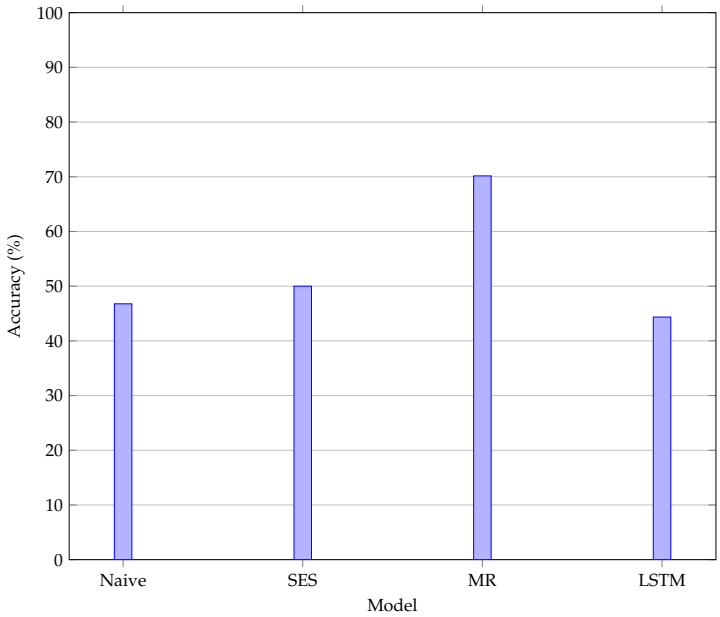


FIGURE A.30: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6.*

A.1.7 Test 7 results

The results obtained for test 7 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.31 shows the predictions made by the LSTM model,
- Figure A.32(a) and Figure A.32(b) show the change in accuracy and error as predictions are made,
- Figure A.33 shows the accuracy obtained for different neural network architectures,
- Figure A.34 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.35 shows the accuracy obtained when predicting the direction of price movement,
- Table A.13 represents the results obtained when the accuracy threshold is changed,
- Table A.14 illustrates the accuracy obtained when the direction of price movement is predicted.

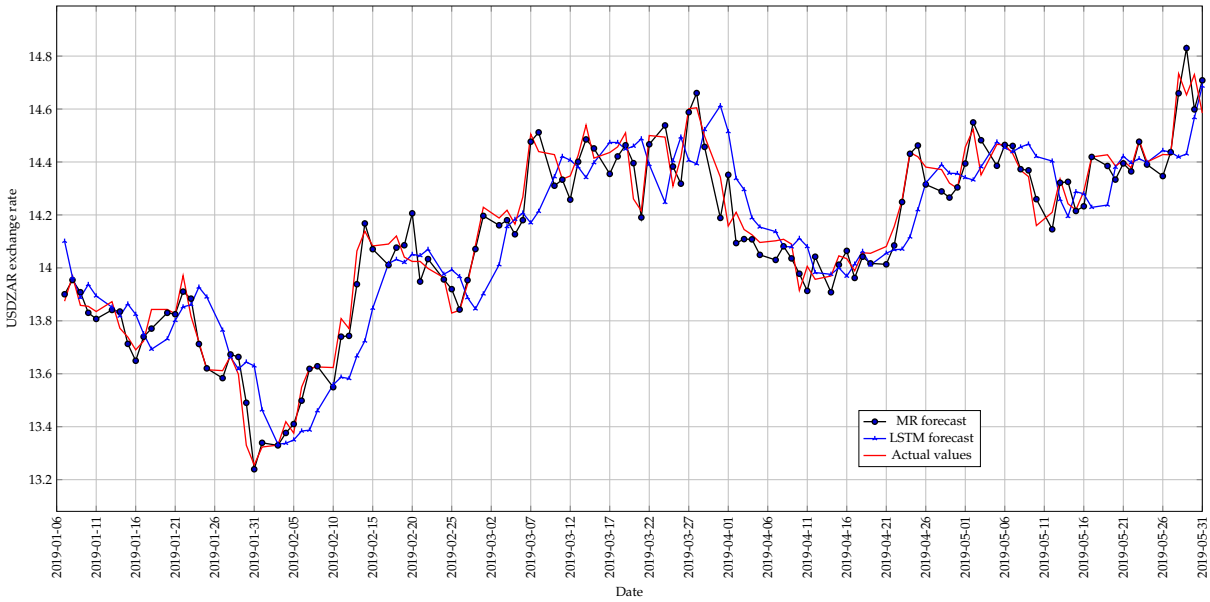
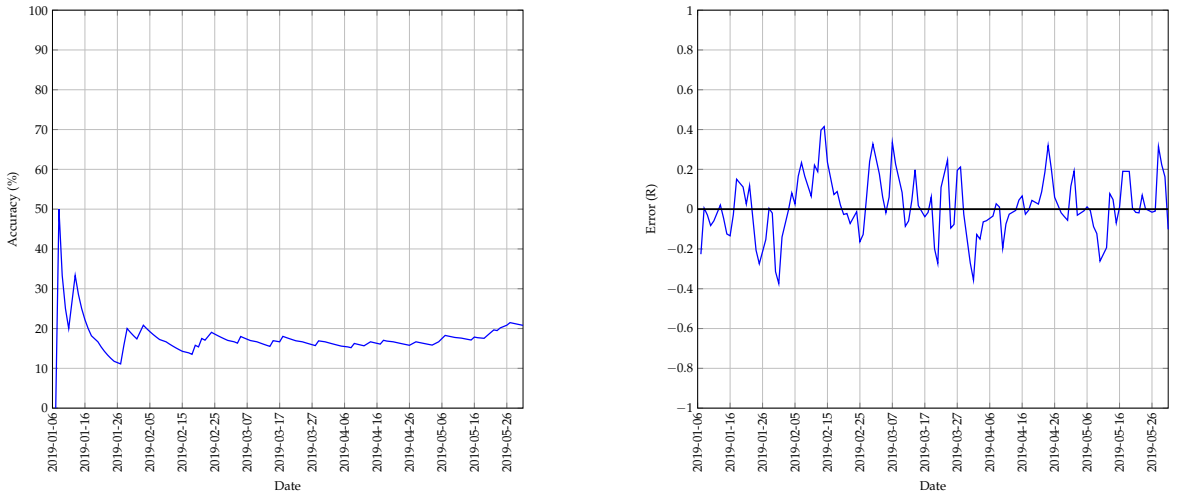


FIGURE A.31: *The best predictions made by the LSTM model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	20.8	36.0	49.6	57.6

TABLE A.13: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.32: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 7.*

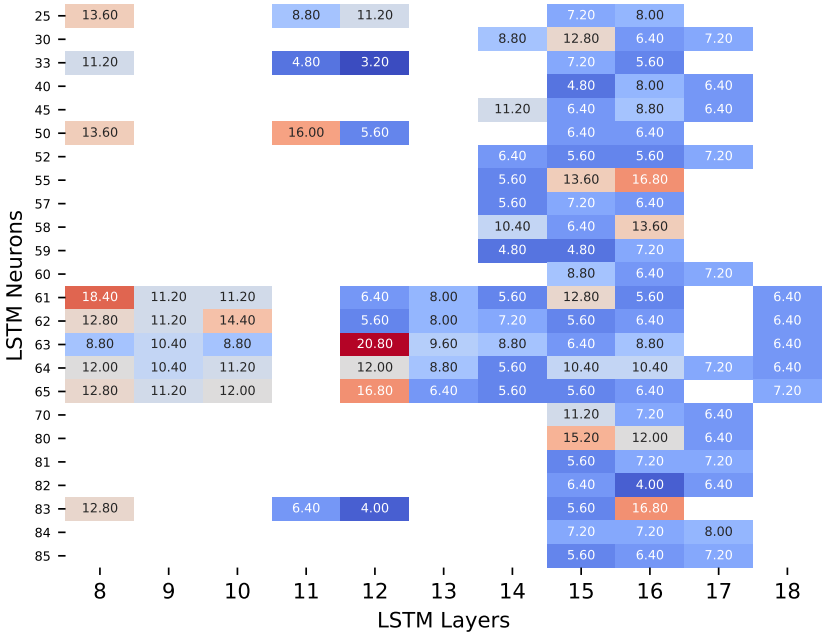


FIGURE A.33: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 7 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	45.161

TABLE A.14: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7.*

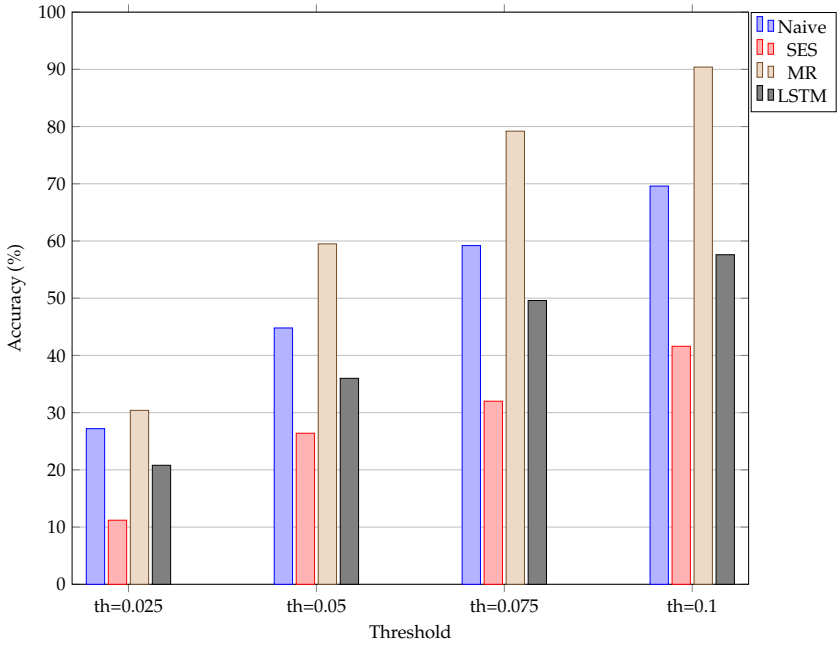


FIGURE A.34: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025.*

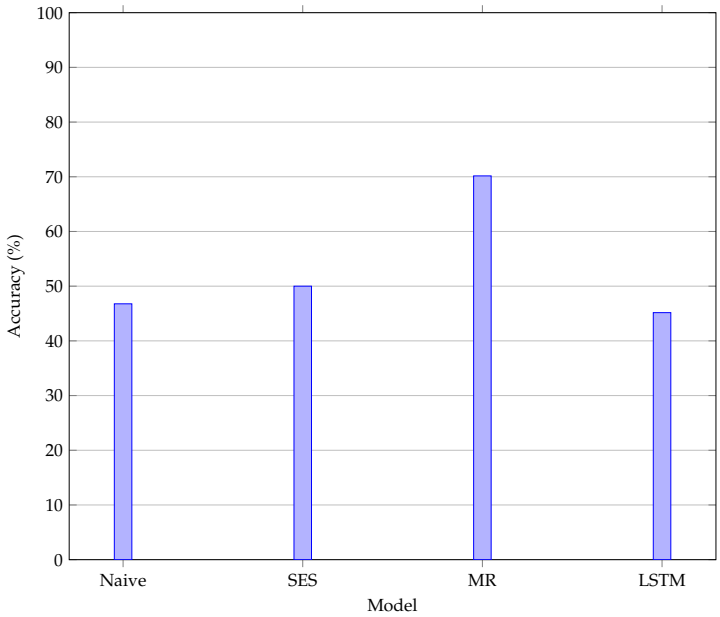


FIGURE A.35: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7.*

A.1.8 Test 8 results

The results obtained for test 8 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.36 shows the predictions made by the LSTM model,
- Figure A.37(a) and Figure A.37(b) show the change in accuracy and error as predictions are made,
- Figure A.38 shows the accuracy obtained for different neural network architectures,
- Figure A.39 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.40 shows the accuracy obtained when predicting the direction of price movement,
- Table A.15 represents the results obtained when the accuracy threshold is changed,
- Table A.16 illustrates the accuracy obtained when the direction of price movement is predicted.

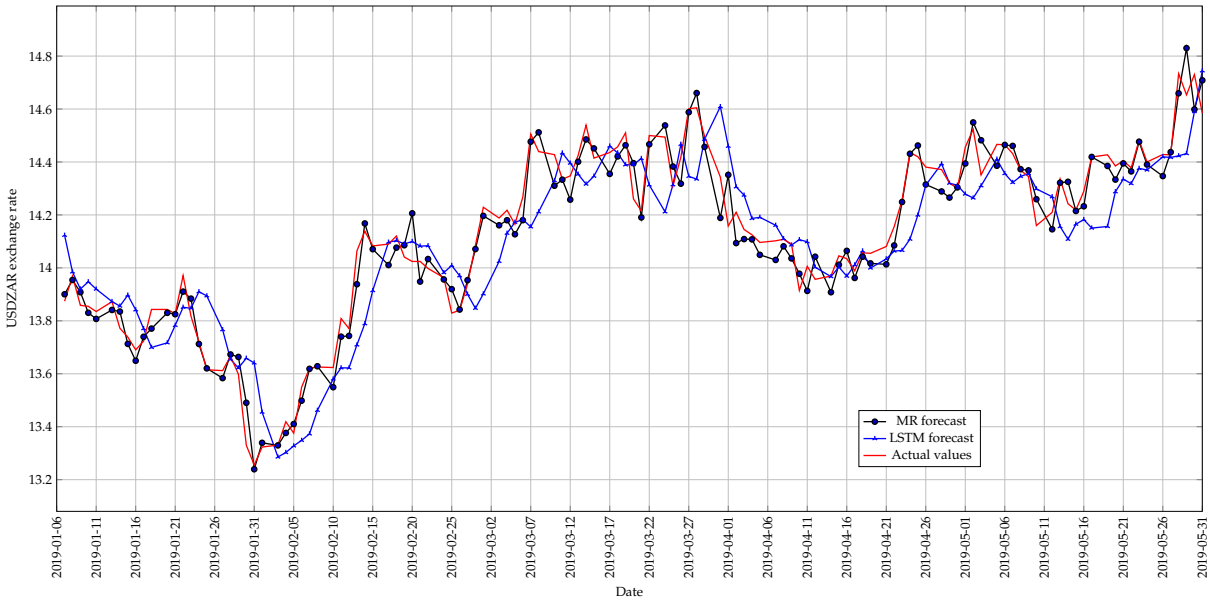
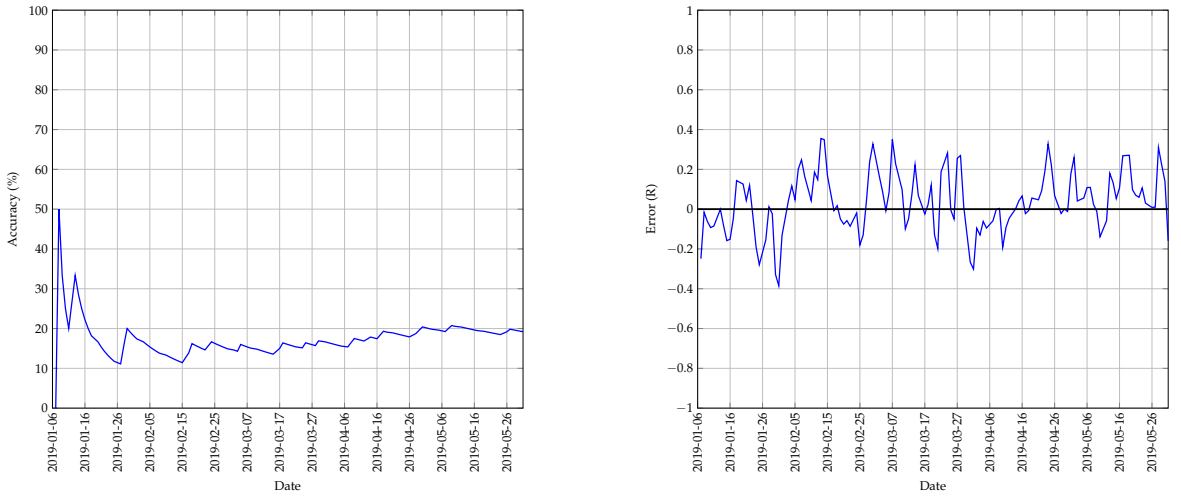


FIGURE A.36: *The best predictions made by the LSTM model for test 8 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	19.2	31.2	42.4	53.6

TABLE A.15: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.37: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 8.*

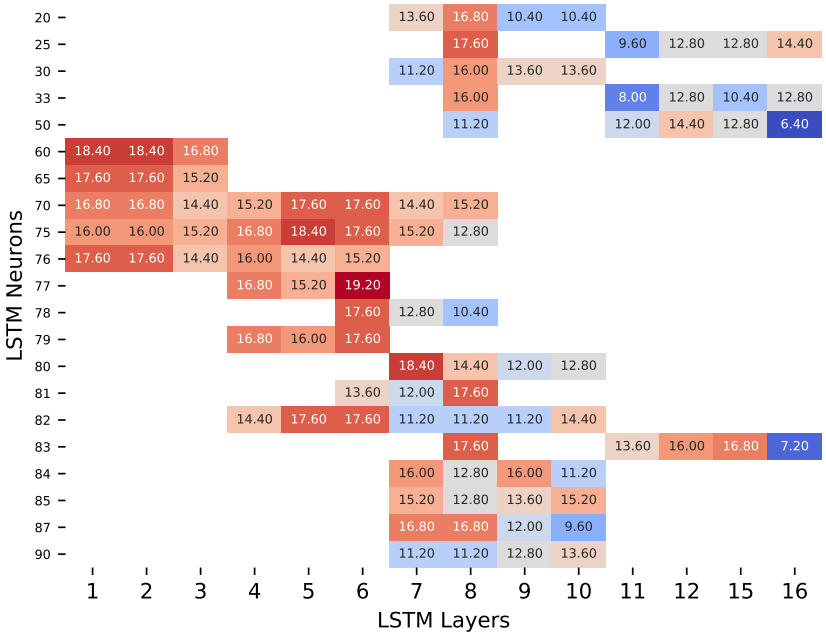


FIGURE A.38: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 8 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	46.774

TABLE A.16: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8.*

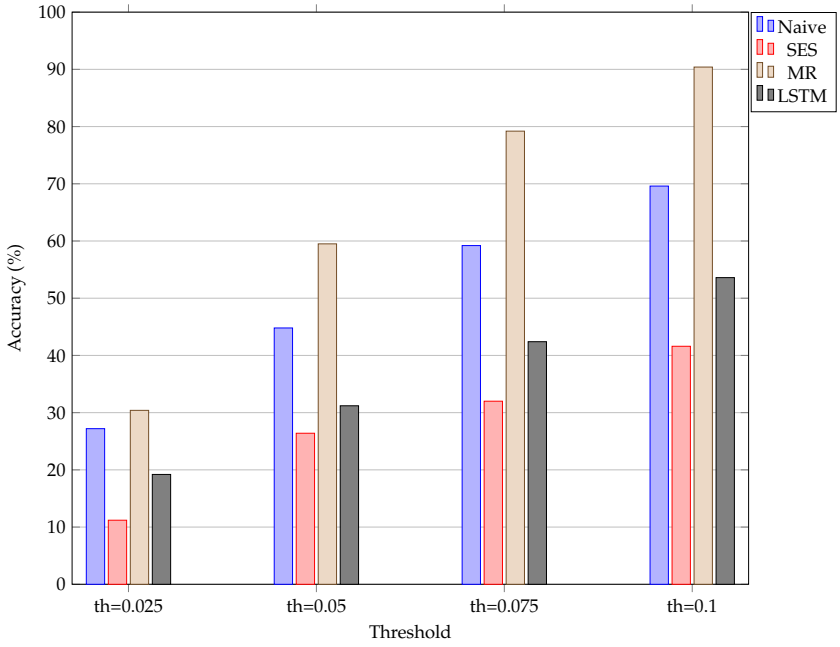


FIGURE A.39: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025.*

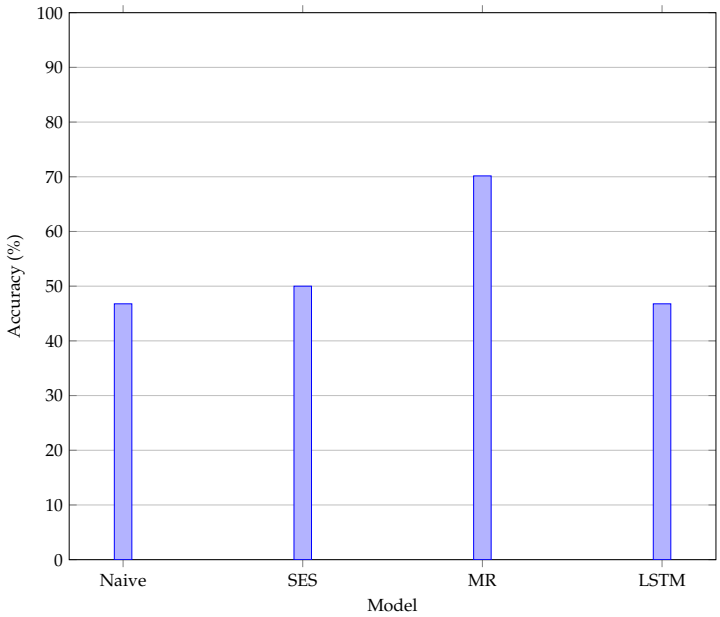


FIGURE A.40: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8.*

A.1.9 Test 9 results

The results obtained for test 9 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.41 shows the predictions made by the LSTM model,
- Figure A.42(a) and Figure A.42(b) show the change in accuracy and error as predictions are made,
- Figure A.43 shows the accuracy obtained for different neural network architectures,
- Figure A.44 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.45 shows the accuracy obtained when predicting the direction of price movement,
- Table A.17 represents the results obtained when the accuracy threshold is changed,
- Table A.18 illustrates the accuracy obtained when the direction of price movement is predicted.

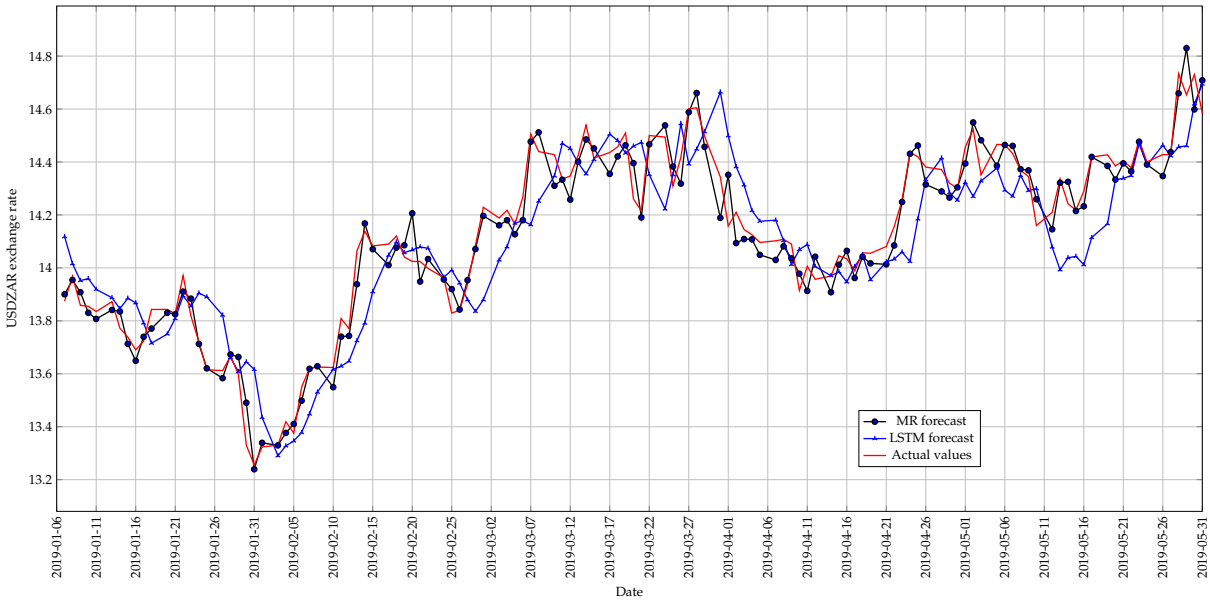
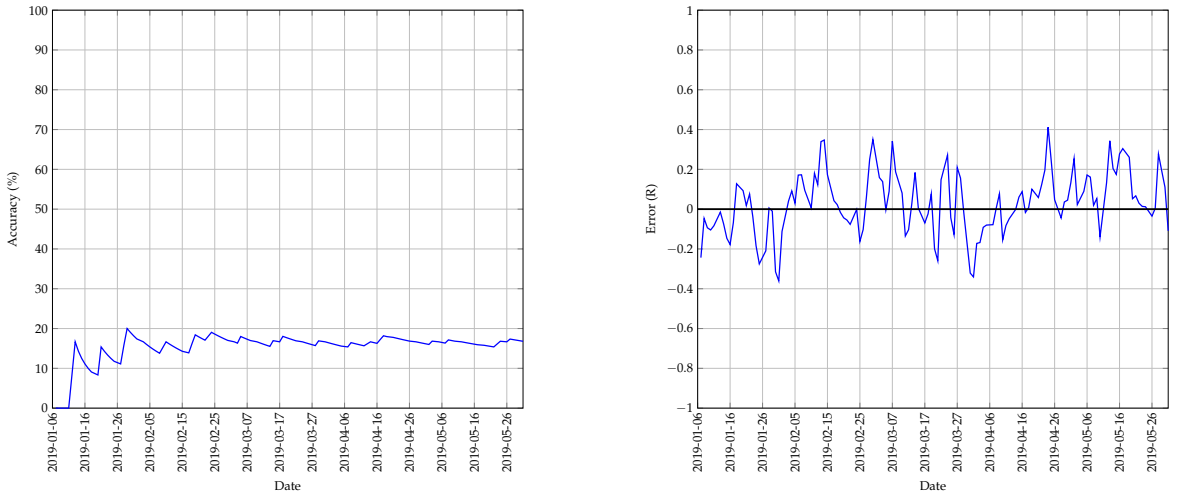


FIGURE A.41: *The best predictions made by the LSTM model for test 9 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	16.8	28.8	36.8	50.4

TABLE A.17: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.42: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 9.*

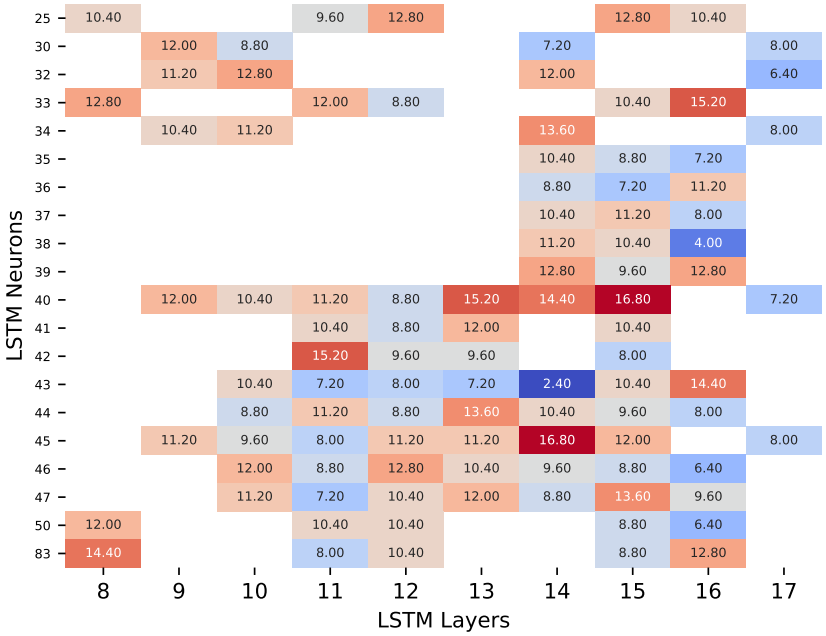


FIGURE A.43: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 9 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	49.194

TABLE A.18: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9.*

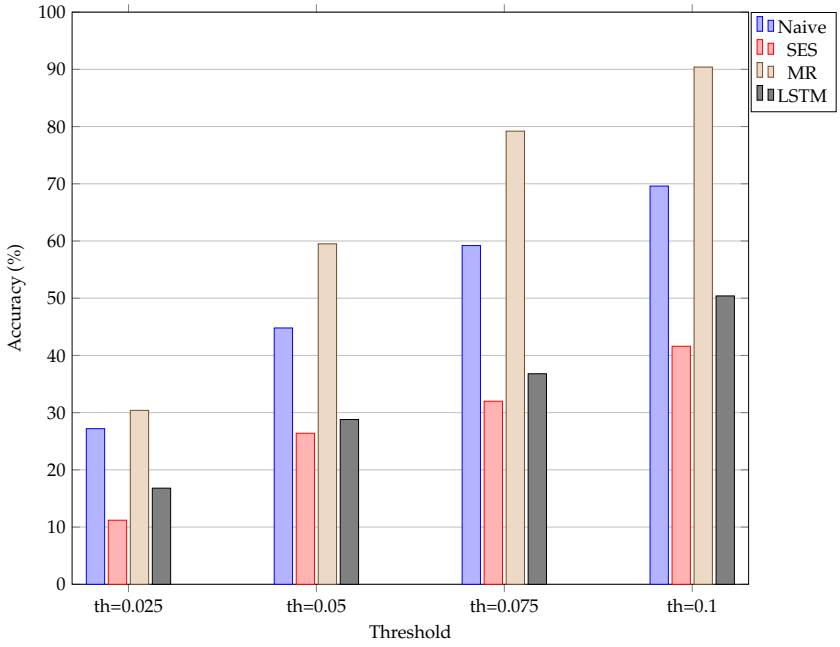


FIGURE A.44: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025.*

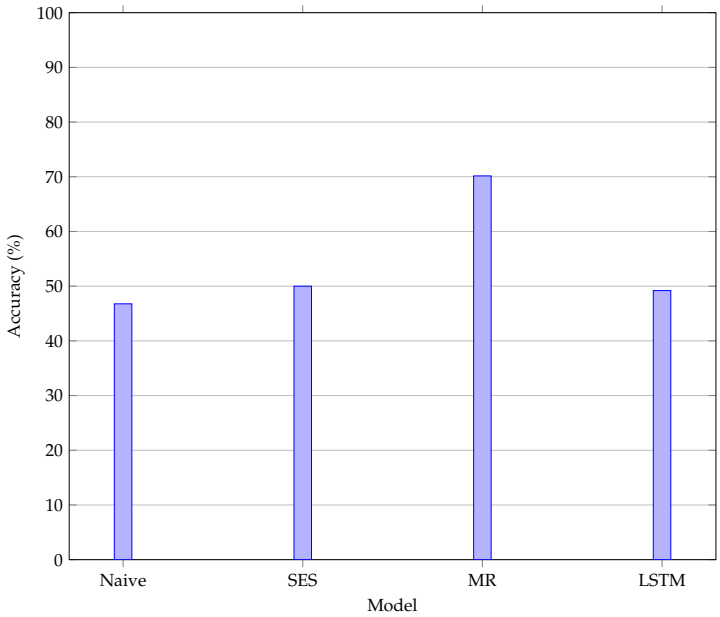


FIGURE A.45: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9.*

A.1.10 Test 10 results

The results obtained for test 10 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.46 shows the predictions made by the LSTM model,
- Figure A.47(a) and Figure A.47(b) show the change in accuracy and error as predictions are made,
- Figure A.48 shows the accuracy obtained for different neural network architectures,
- Figure A.49 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.50 shows the accuracy obtained when predicting the direction of price movement,
- Table A.19 represents the results obtained when the accuracy threshold is changed,
- Table A.20 illustrates the accuracy obtained when the direction of price movement is predicted.

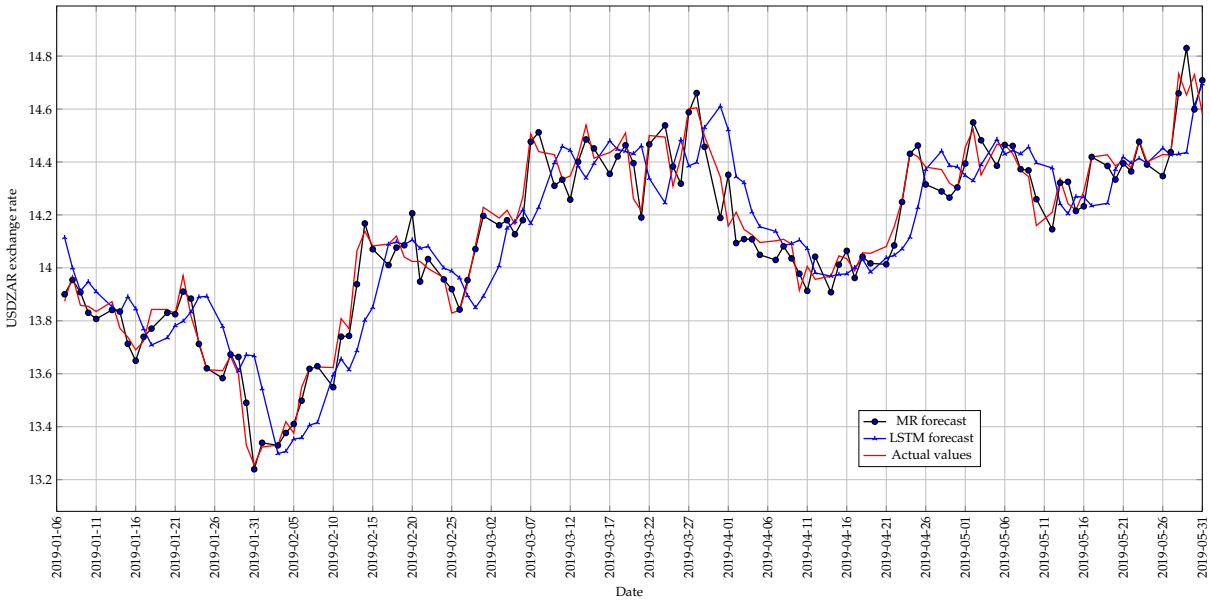
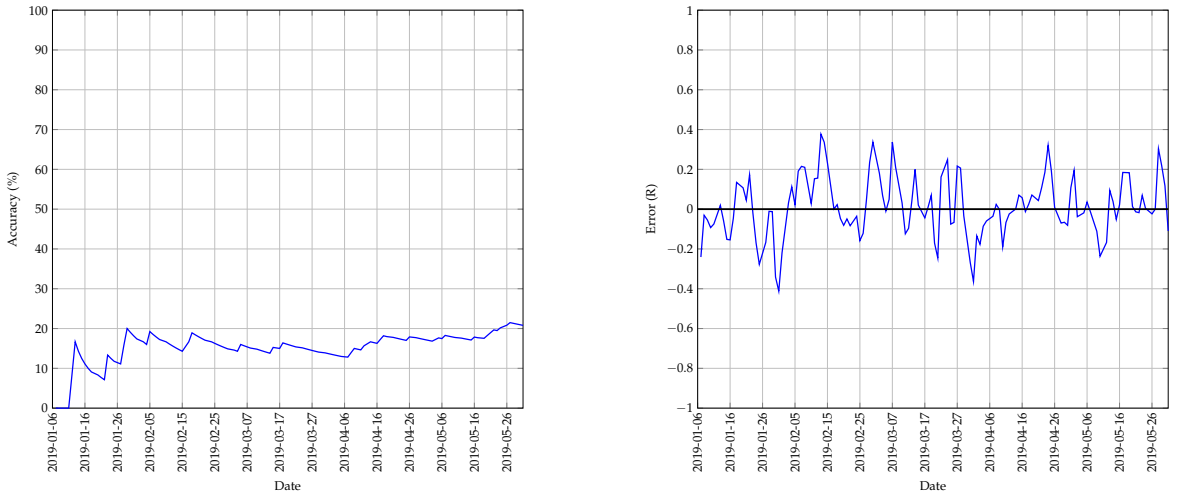


FIGURE A.46: *The best predictions made by the LSTM model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	20.8	36.0	48.8	55.2

TABLE A.19: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.47: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 10.*

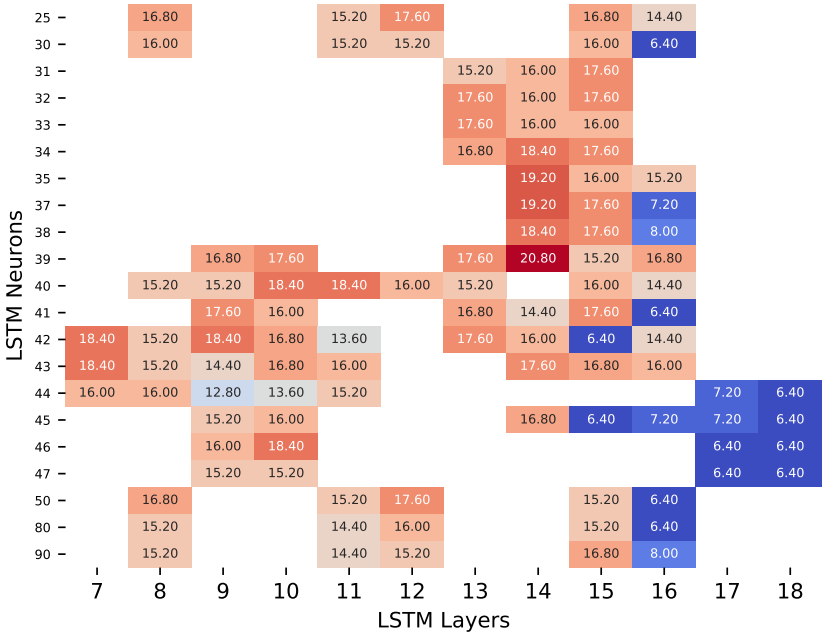


FIGURE A.48: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 10 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	45.968

TABLE A.20: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10.*

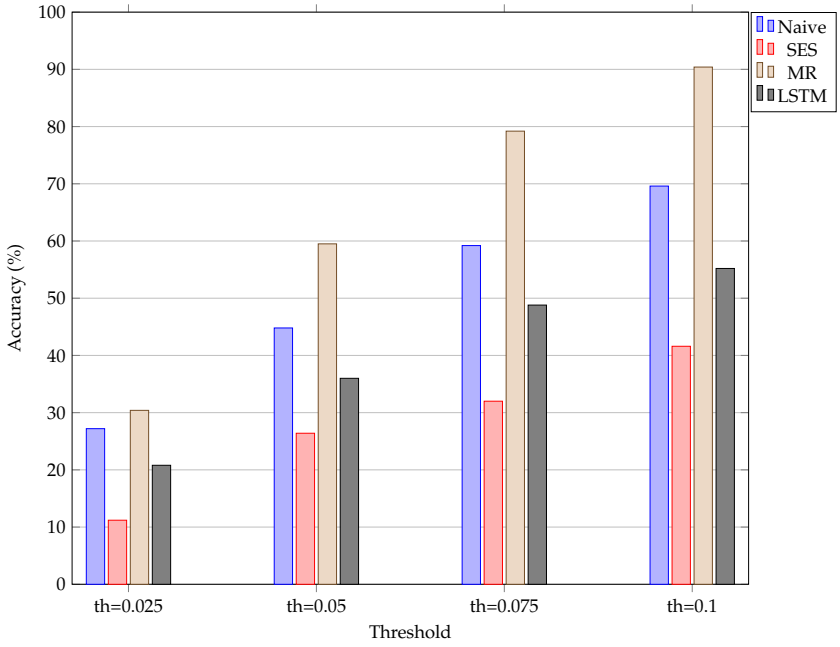


FIGURE A.49: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025.*

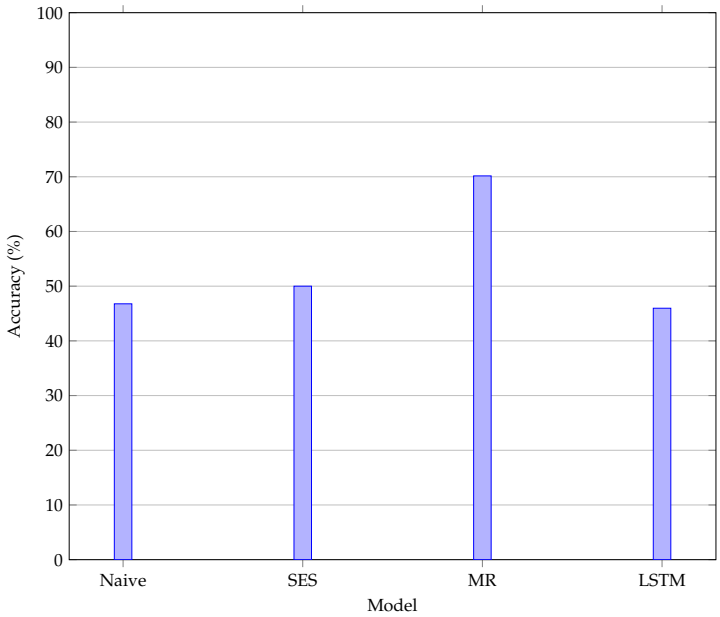


FIGURE A.50: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10.*

A.1.11 Test 11 results

The results obtained for test 11 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.51 shows the predictions made by the LSTM model,
- Figure A.52(a) and Figure A.52(b) show the change in accuracy and error as predictions are made,
- Figure A.53 shows the accuracy obtained for different neural network architectures,
- Figure A.54 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.55 shows the accuracy obtained when predicting the direction of price movement,
- Table A.21 represents the results obtained when the accuracy threshold is changed,
- Table A.22 illustrates the accuracy obtained when the direction of price movement is predicted.

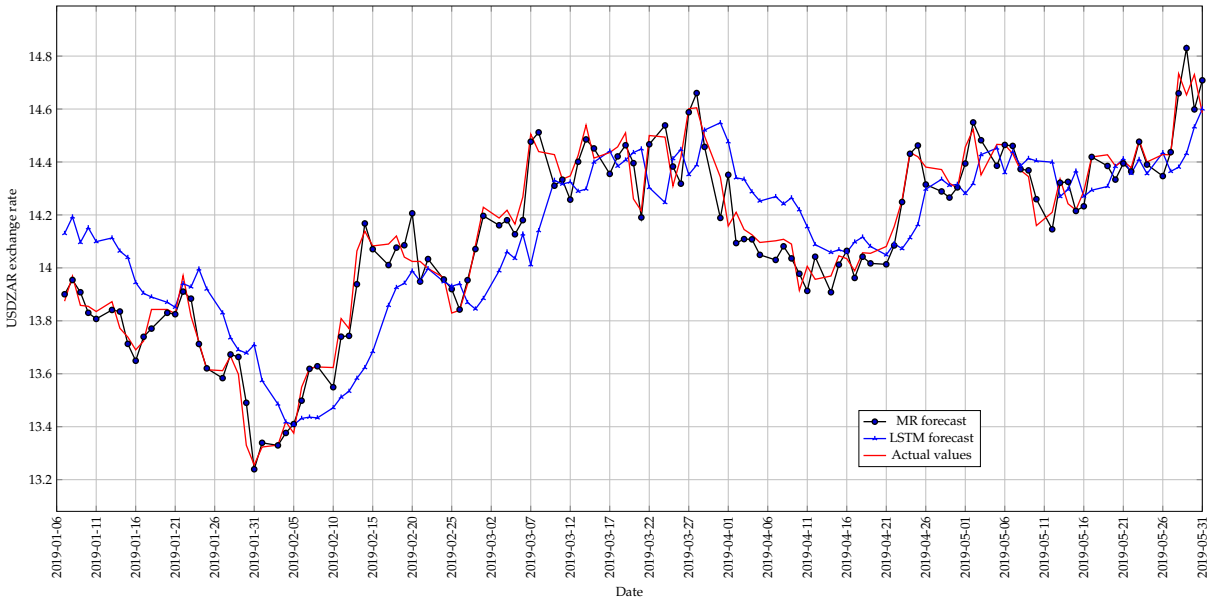
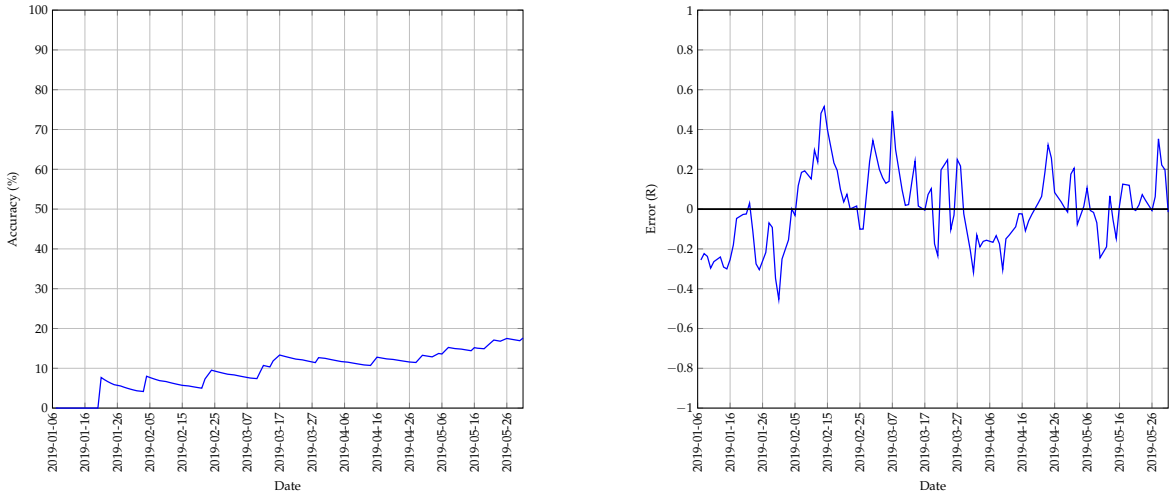


FIGURE A.51: *The best predictions made by the LSTM model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	17.6	25.6	34.4	39.2

TABLE A.21: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.52: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 11.*

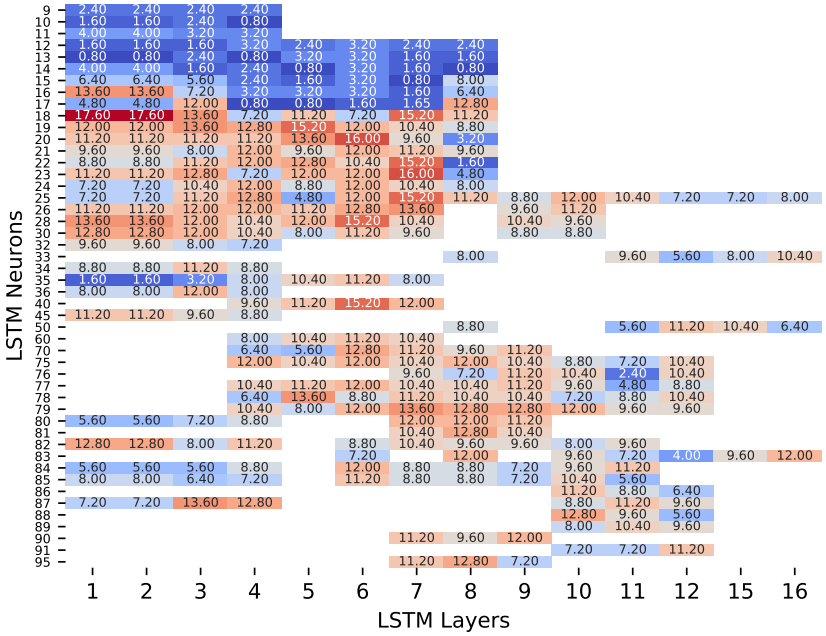


FIGURE A.53: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 11 with 1 previous day as input.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
LSTM	54.032

TABLE A.22: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11.*

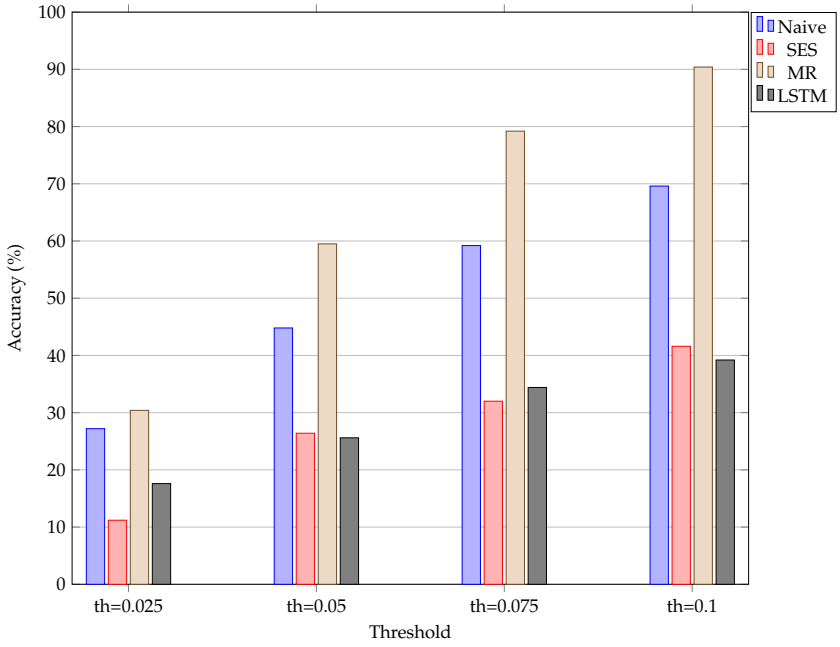


FIGURE A.54: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025.*

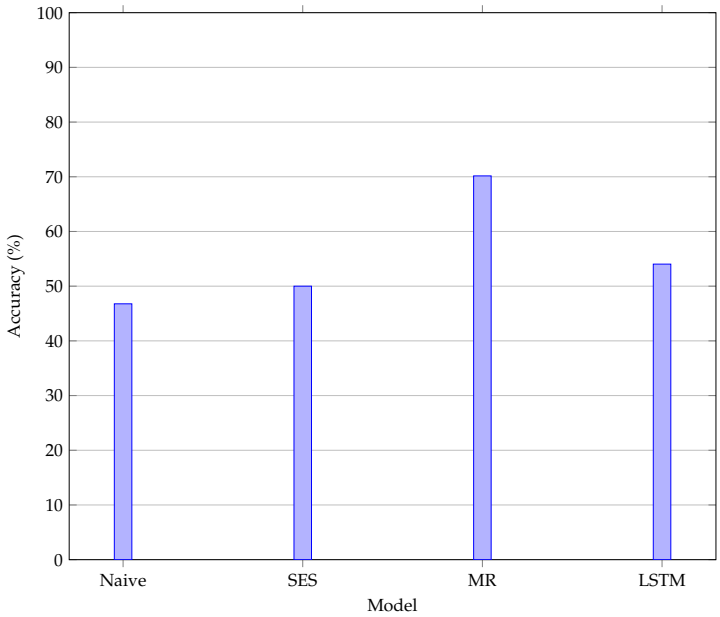


FIGURE A.55: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11.*

A.1.12 Test 12 results

The results obtained for test 12 of the LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.56 shows the predictions made by the LSTM model,
- Figure A.57(a) and Figure A.57(b) show the change in accuracy and error as predictions are made,
- Figure A.58 shows the accuracy obtained for different neural network architectures,
- Figure A.59 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.60 shows the accuracy obtained when predicting the direction of price movement,
- Table A.23 represents the results obtained when the accuracy threshold is changed,
- Table A.24 illustrates the accuracy obtained when the direction of price movement is predicted.

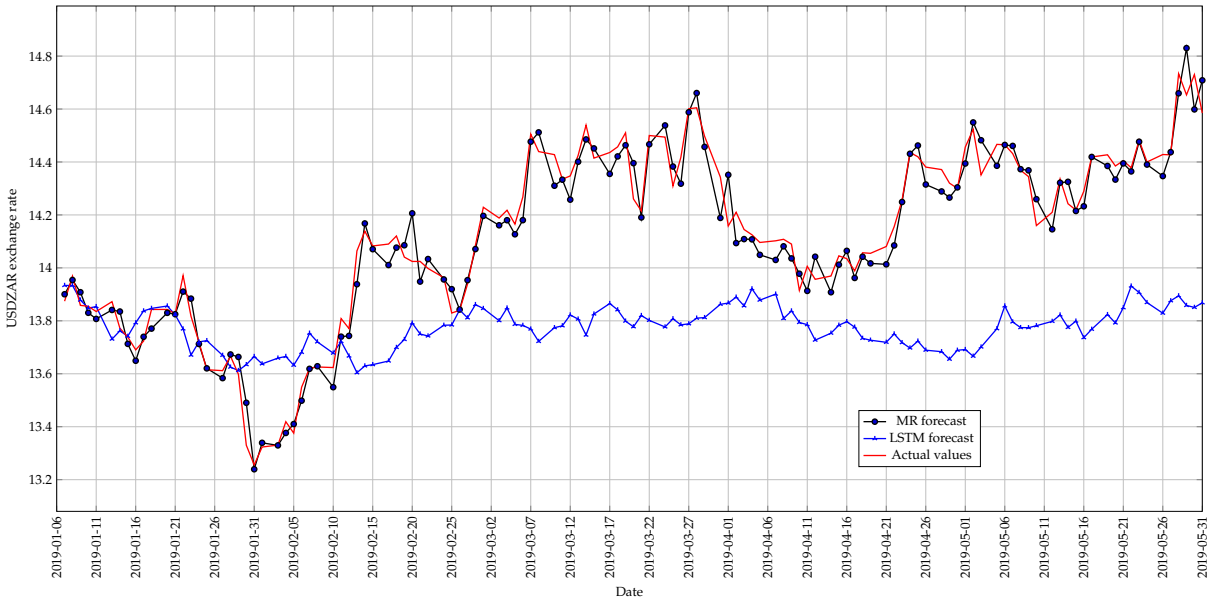
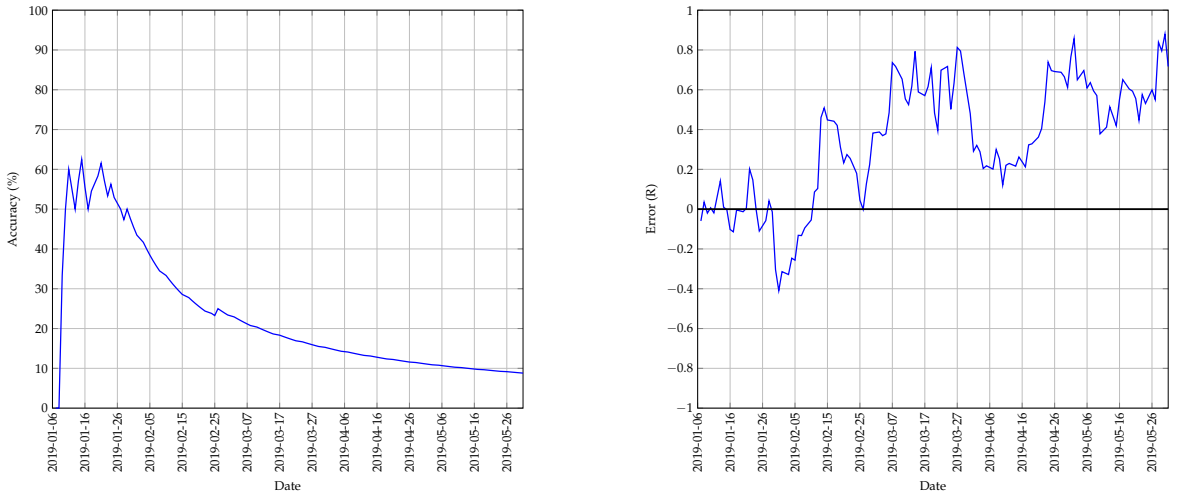


FIGURE A.56: *The best predictions made by the LSTM model for test 12 with 1 previous day as input using a walk-forward validation approach on the test set.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
LSTM	8.8	11.2	13.6	15.2

TABLE A.23: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12.*



(a) *The change in accuracy of the LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.57: *A graphical representation of how the accuracy and error of the LSTM model changed whilst making predictions on the test set for test 12.*

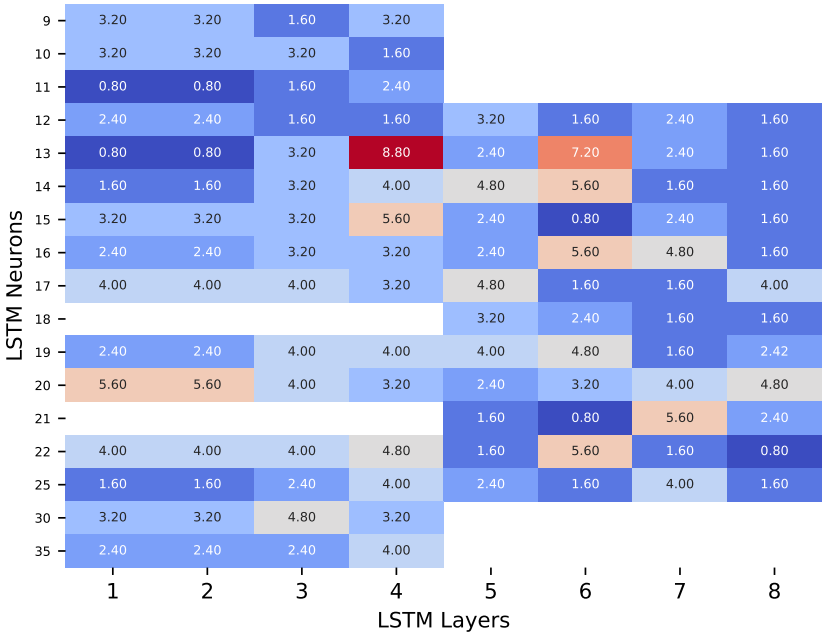


FIGURE A.58: *A heatmap that illustrates the best accuracy obtained for different architectures of the LSTM model for test 12 with 5 previous day as input.*

Model	Accuracy (%)
Naive	47.581
SES	50.000
MR	70.161
LSTM	50.0

TABLE A.24: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12.*

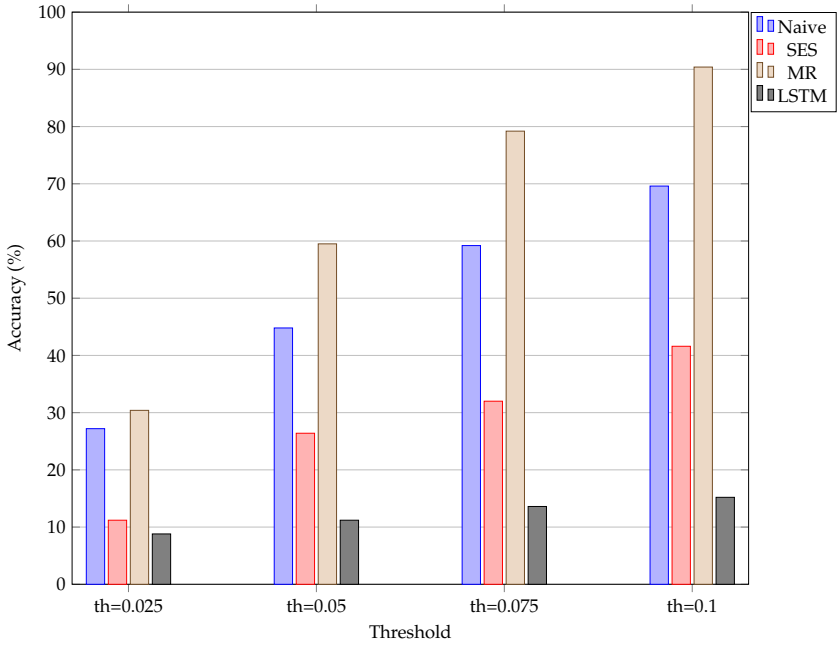


FIGURE A.59: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025.*

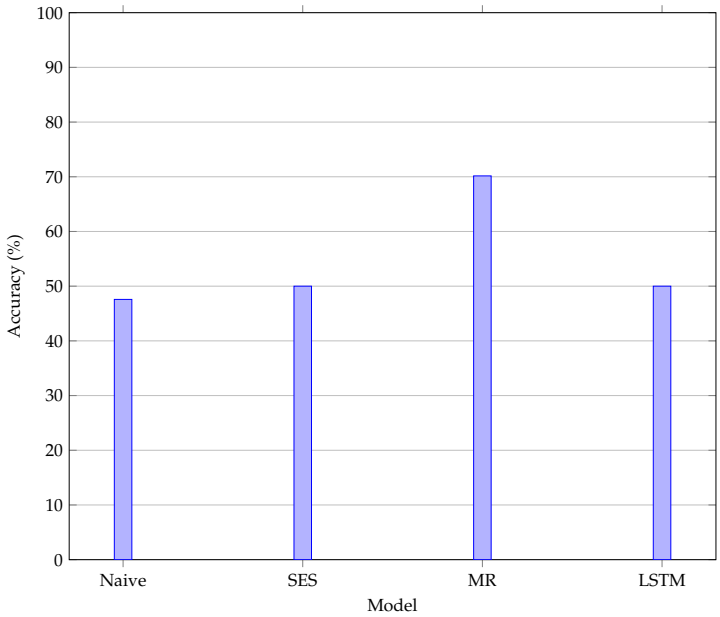


FIGURE A.60: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12.*

A.2 CNN-LSTM model results

This section contains the results obtained by the CNN-LSTM model.

A.2.1 Test 1 results

The results obtained for test 1 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.61 shows the predictions made by the CNN-LSTM model,
- Figure A.62(a) and Figure A.62(b) show the change in accuracy and error as predictions are made,
- Figure A.63(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.63(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.64 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.65 shows the accuracy obtained when predicting the direction of price movement,
- Table A.25 represents the results obtained when the accuracy threshold is changed,
- Table A.26 illustrates the accuracy obtained when the direction of price movement is predicted.

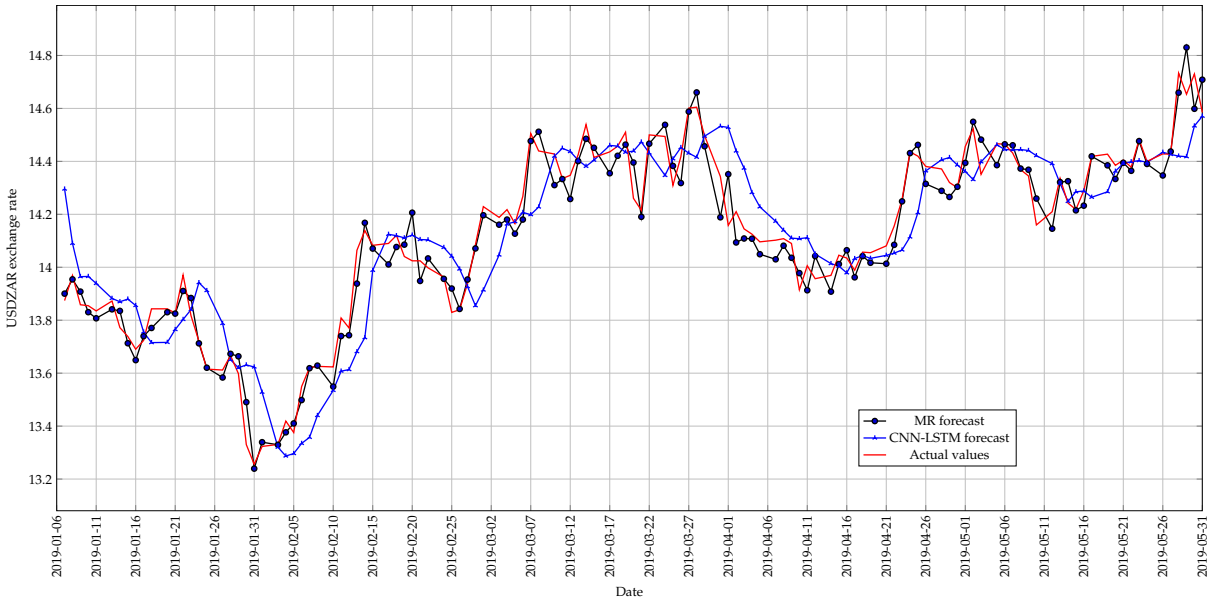
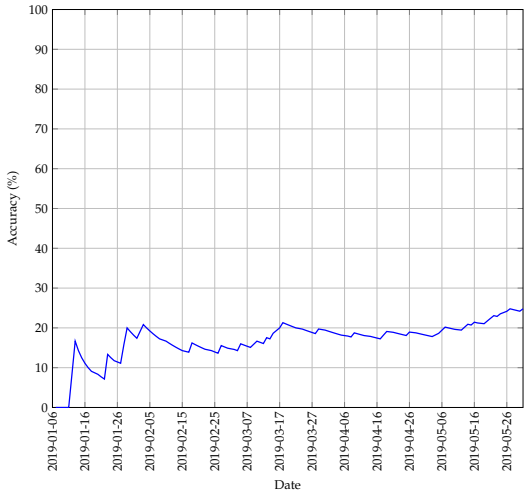
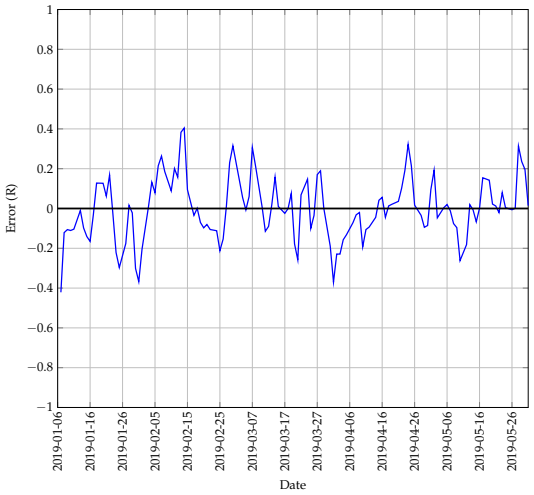


FIGURE A.61: *The best predictions made by the CNN-LSTM model for test 1 with 2 previous day as input using a walk-forward validation approach on the test set.*

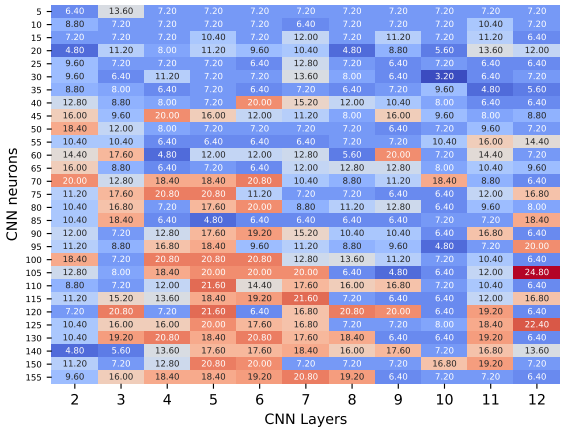


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

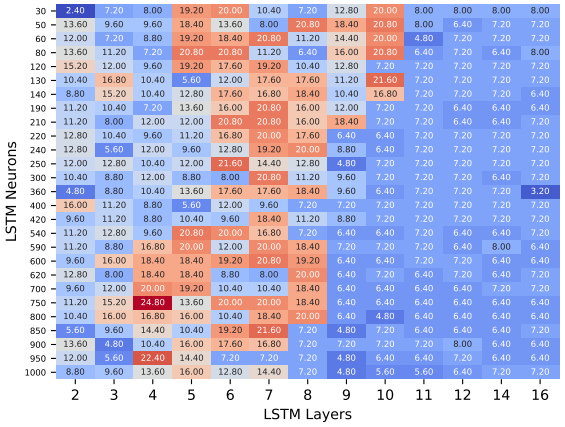


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.62: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 1.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 2 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 2 previous days as input.*

FIGURE A.63: *A graphical representation of the influence that different architectures have on prediction accuracy for test 1 of the CNN-LSTM model.*

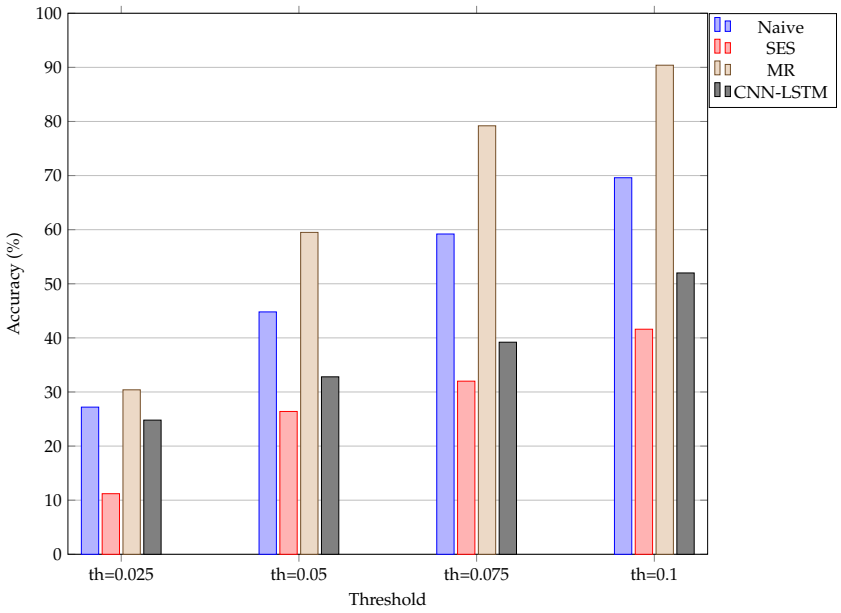


FIGURE A.64: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	24.8	32.8	39.2	52.0

TABLE A.25: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	52.419

TABLE A.26: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1.*

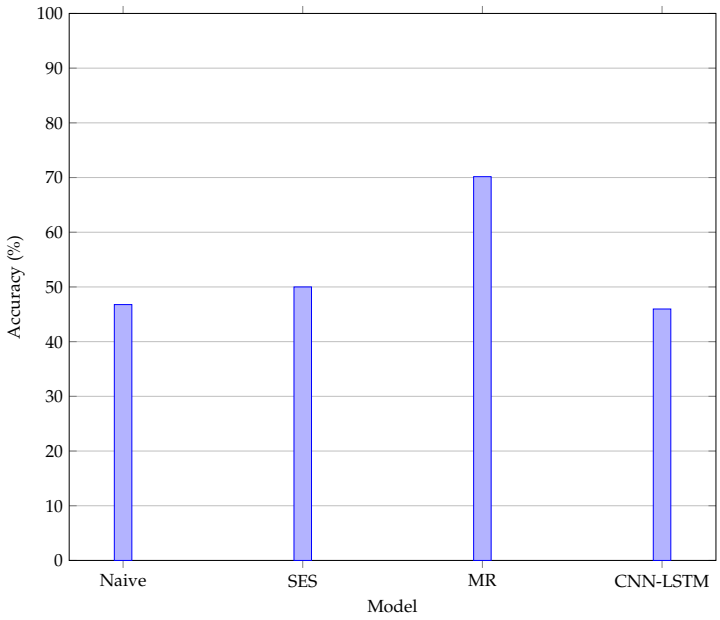


FIGURE A.65: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1.*

A.2.2 Test 2 results

The results obtained for test 2 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.66 shows the predictions made by the CNN-LSTM model,
- Figure A.67(a) and Figure A.67(b) show the change in accuracy and error as predictions are made,
- Figure A.68(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.68(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.69 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.70 shows the accuracy obtained when predicting the direction of price movement,
- Table A.27 represents the results obtained when the accuracy threshold is changed,
- Table A.28 illustrates the accuracy obtained when the direction of price movement is predicted.

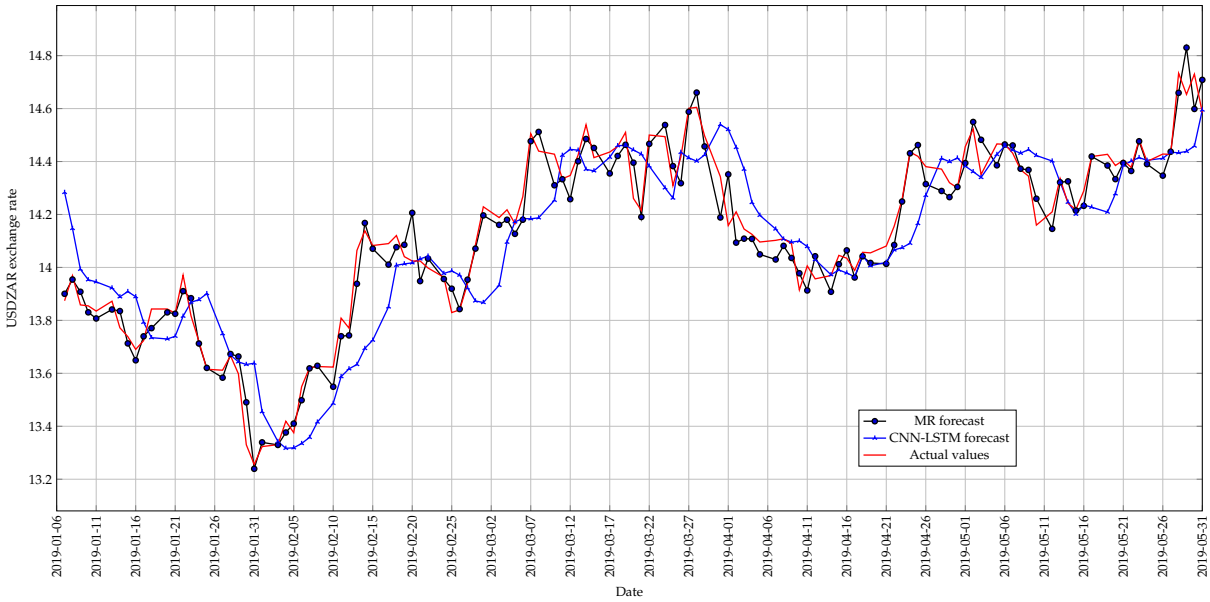
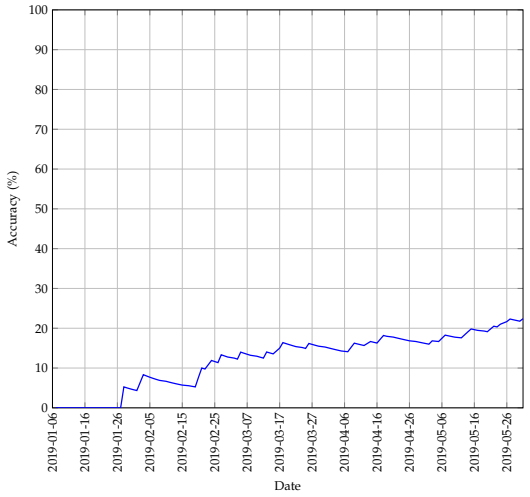
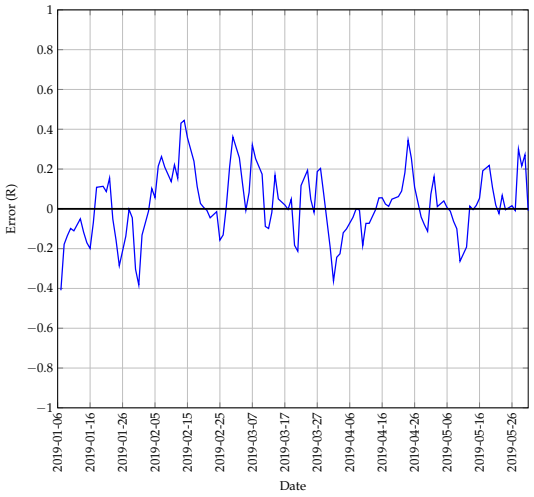


FIGURE A.66: *The best predictions made by the CNN-LSTM model for test 2 with 3 previous days as input using a walk-forward validation approach on the test set.*

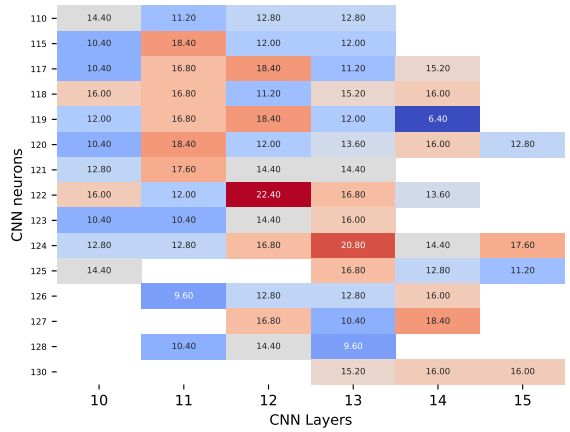


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

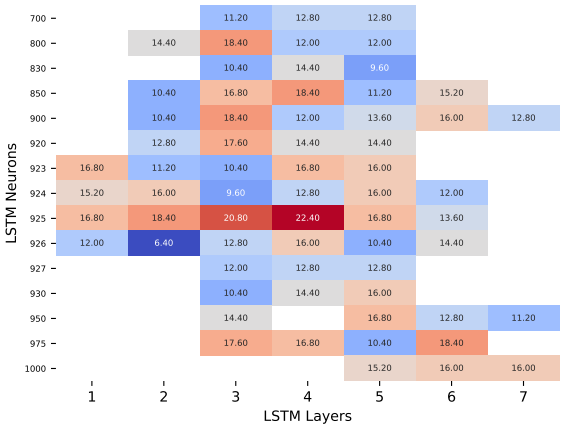


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.67: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 2.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 1 previous day as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 1 previous day as input.*

FIGURE A.68: *A graphical representation of the influence that different architectures have on prediction accuracy for test 2 of the CNN-LSTM model.*

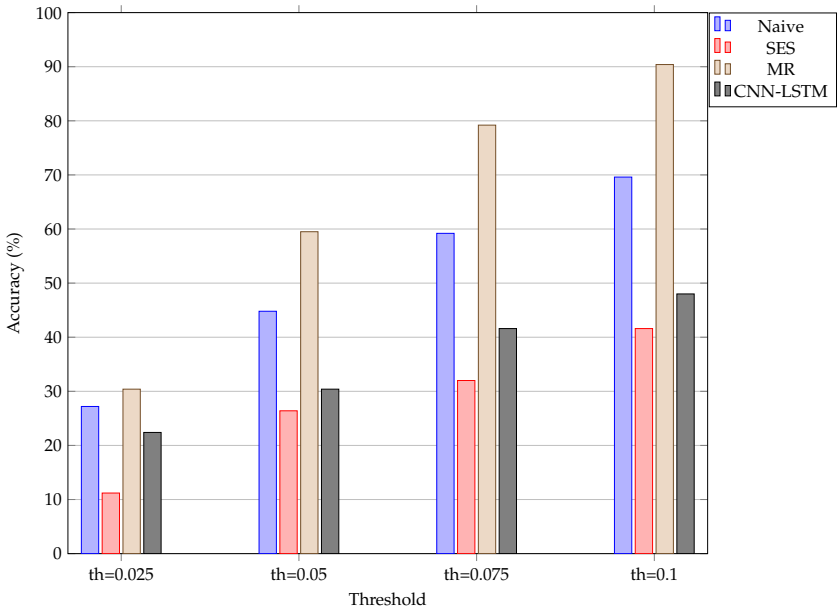


FIGURE A.69: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	22.4	30.4	41.6	48.0

TABLE A.27: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2.*

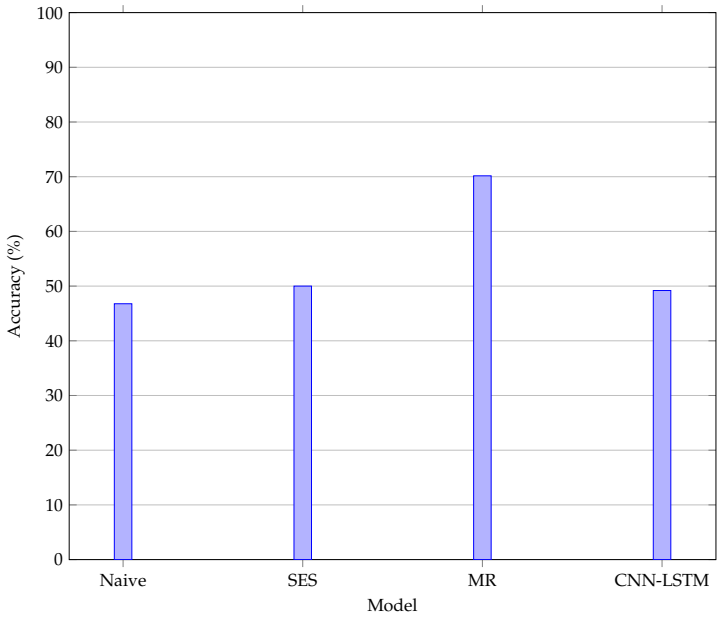


FIGURE A.70: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	52.419

TABLE A.28: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2.*

A.2.3 Test 3 results

The results obtained for test 3 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.71 shows the predictions made by the CNN-LSTM model,
- Figure A.72(a) and Figure A.72(b) show the change in accuracy and error as predictions are made,
- Figure A.73(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.73(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.74 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.75 shows the accuracy obtained when predicting the direction of price movement,
- Table A.29 represents the results obtained when the accuracy threshold is changed,
- Table A.30 illustrates the accuracy obtained when the direction of price movement is predicted.

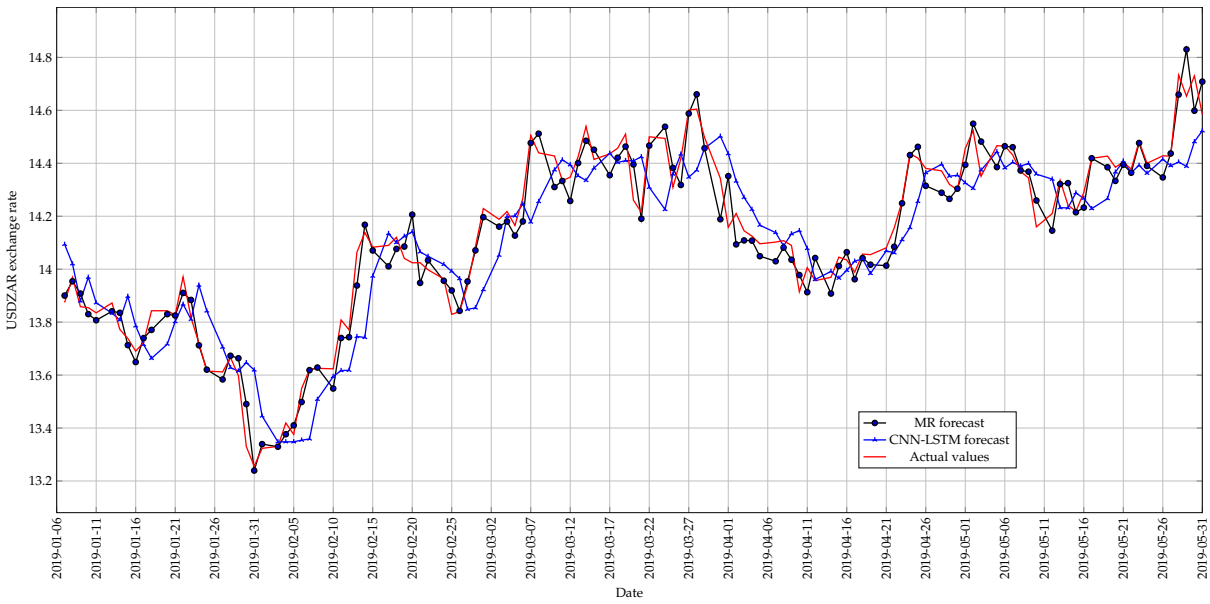
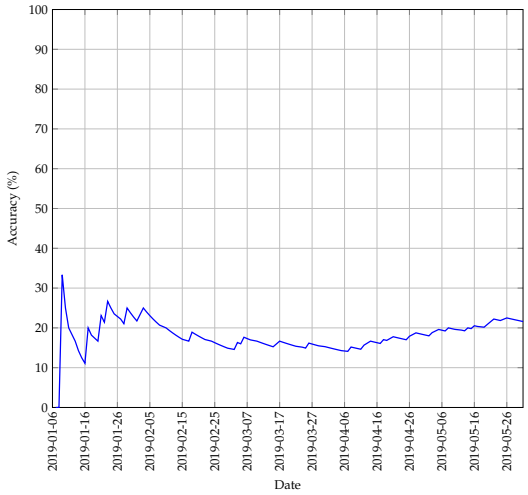
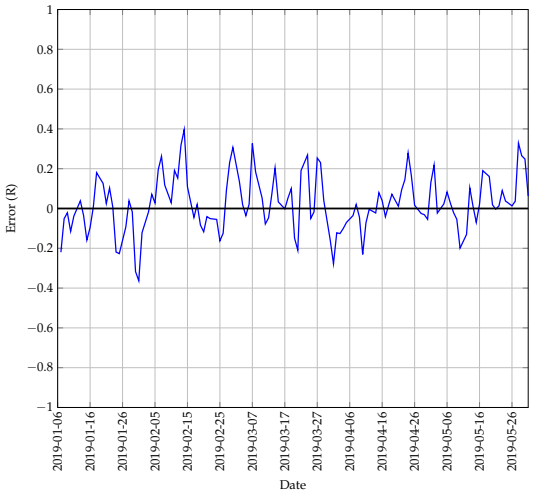


FIGURE A.71: *The best predictions made by the CNN-LSTM model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set.*

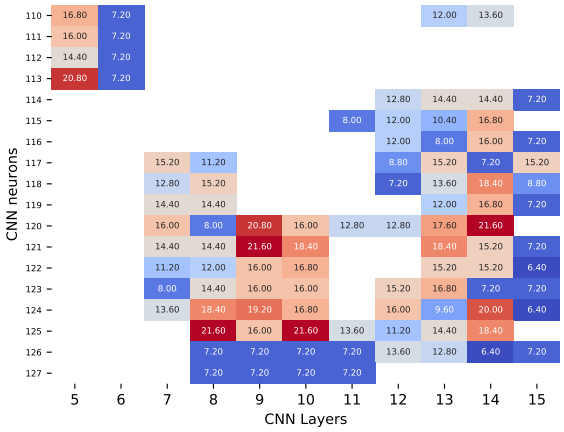


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

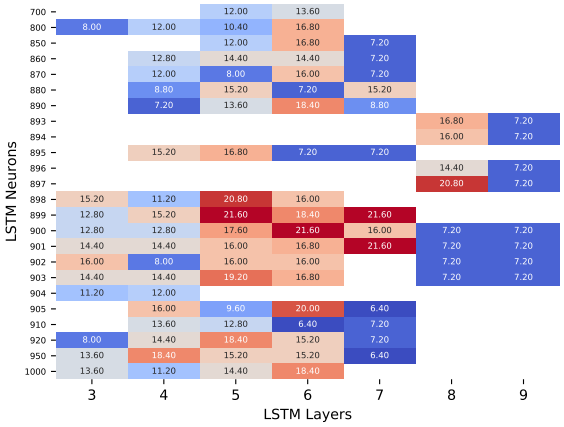


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.72: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 3.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 1 previous day as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 1 previous day as input.*

FIGURE A.73: *A graphical representation of the influence that different architectures have on prediction accuracy for test 3 of the CNN-LSTM model.*

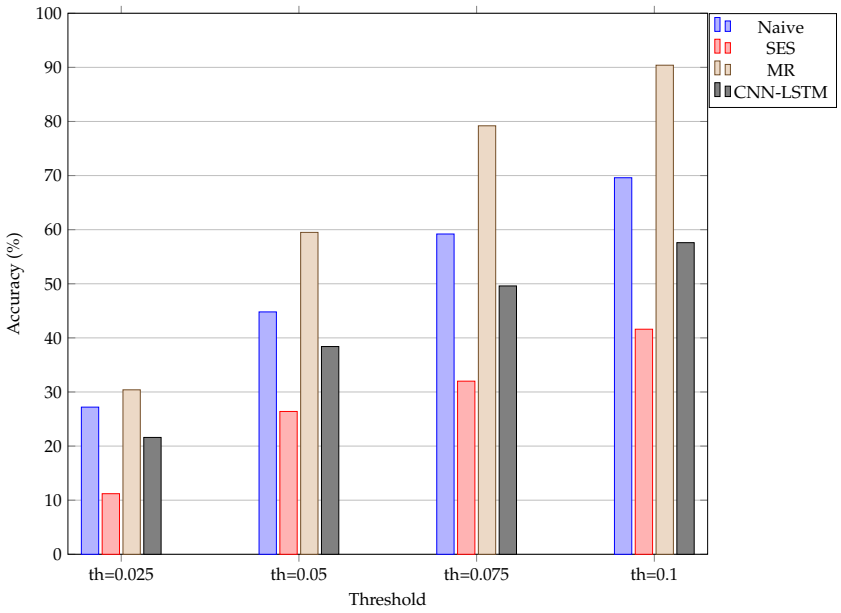


FIGURE A.74: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	21.6	38.4	49.6	57.6

TABLE A.29: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	43.548

TABLE A.30: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3.*

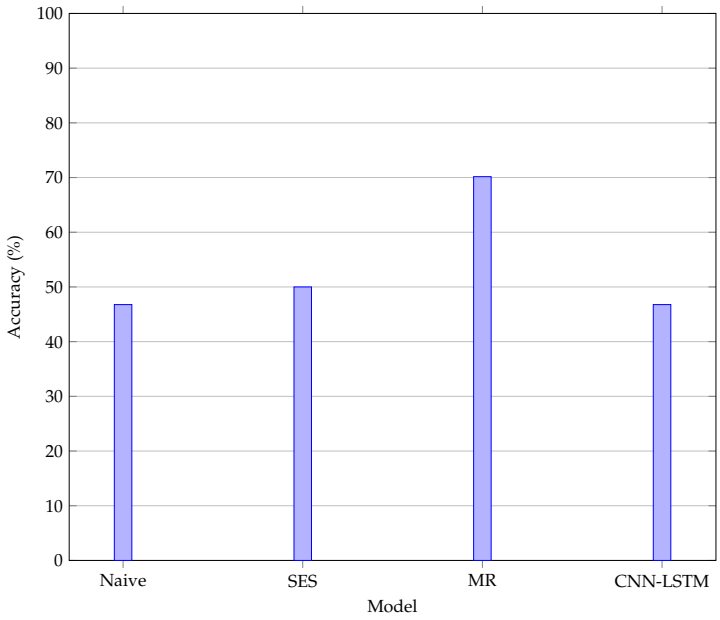


FIGURE A.75: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3.*

A.2.4 Test 4 results

The results obtained for test 4 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.76 shows the predictions made by the CNN-LSTM model,
- Figure A.77(a) and Figure A.77(b) show the change in accuracy and error as predictions are made,
- Figure A.78(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.78(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.79 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.80 shows the accuracy obtained when predicting the direction of price movement,
- Table A.31 represents the results obtained when the accuracy threshold is changed,
- Table A.32 illustrates the accuracy obtained when the direction of price movement is predicted.

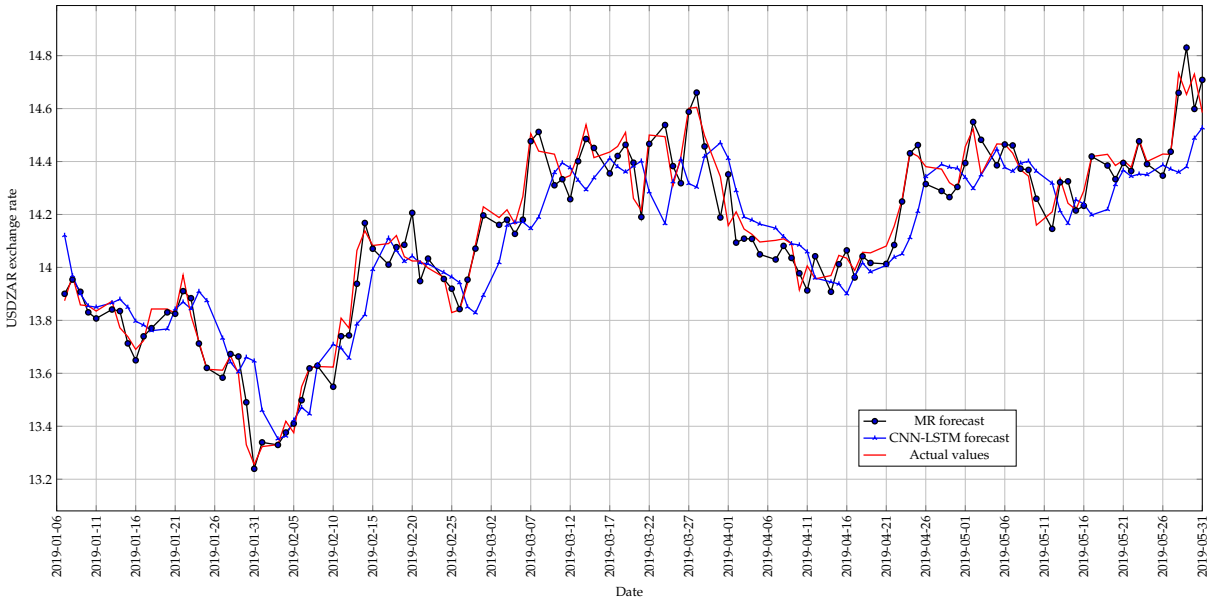
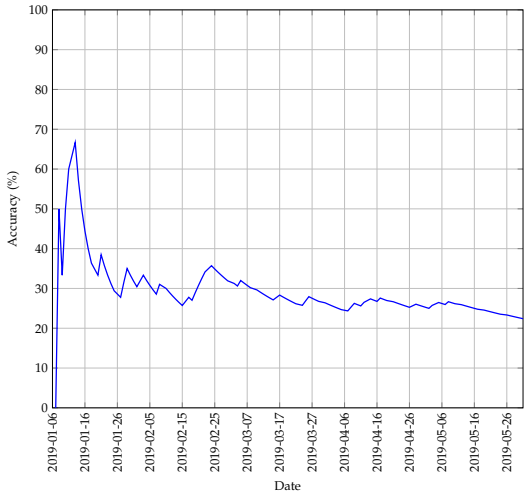
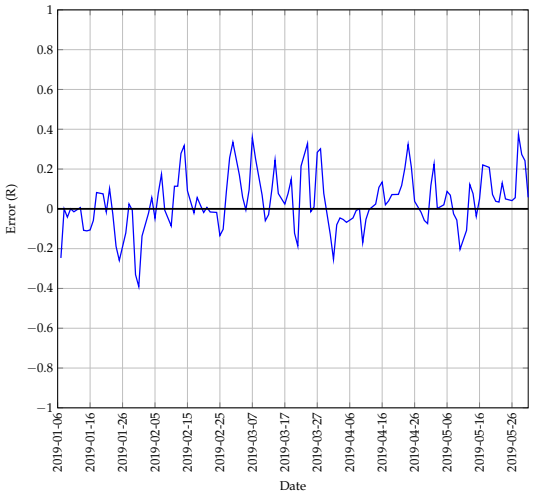


FIGURE A.76: *The best predictions made by the CNN-LSTM model for test 4 with 2 previous days as input using a walk-forward validation approach on the test set.*

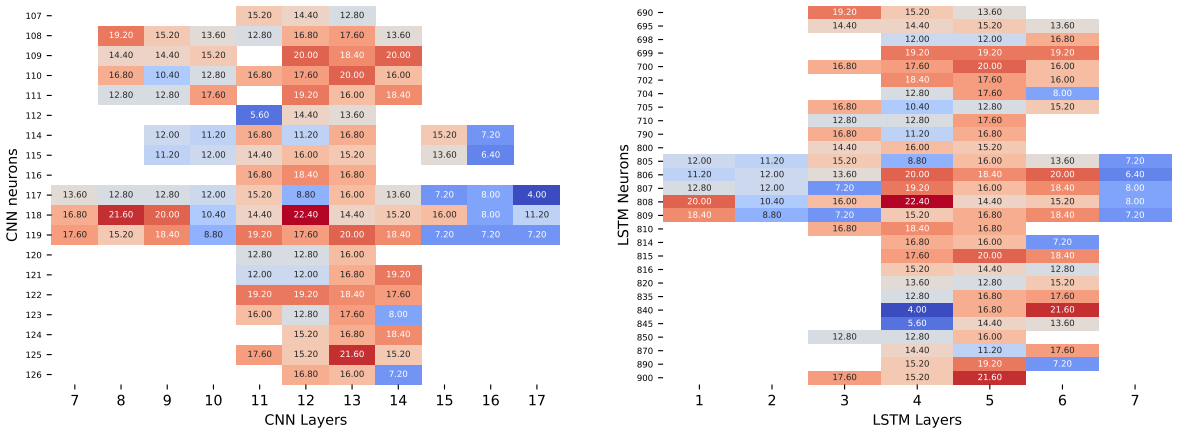


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.77: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 4.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 2 previous days as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 2 previous days as input.*

FIGURE A.78: *A graphical representation of the influence that different architectures have on prediction accuracy for test 4 of the CNN-LSTM model.*

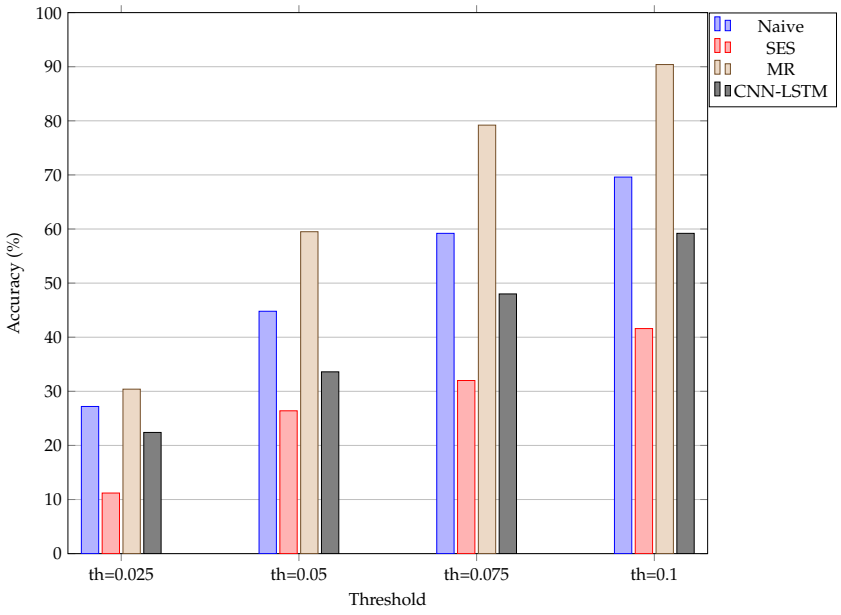


FIGURE A.79: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	22.4	33.6	48.0	59.2

TABLE A.31: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4.*

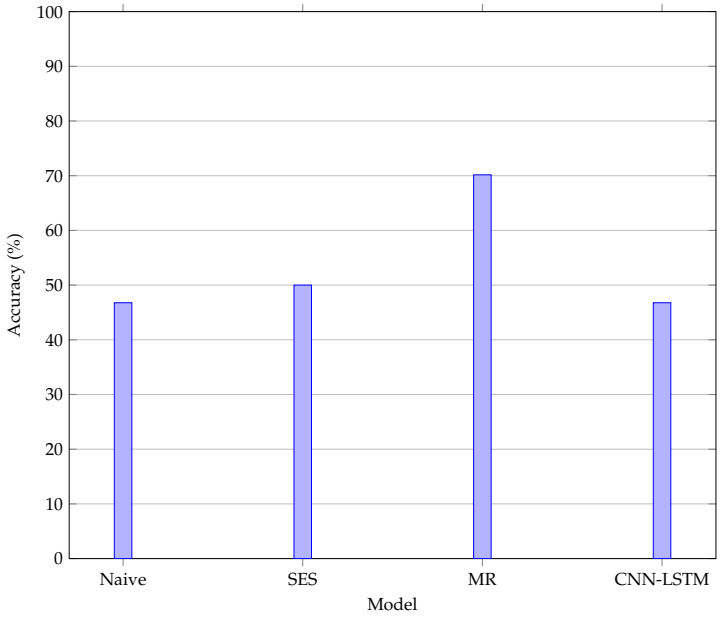


FIGURE A.80: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	46.774

TABLE A.32: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4.*

A.2.5 Test 5 results

The results obtained for test 5 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.81 shows the predictions made by the CNN-LSTM model,
- Figure A.82(a) and Figure A.82(b) show the change in accuracy and error as predictions are made,
- Figure A.83(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.83(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.84 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.85 shows the accuracy obtained when predicting the direction of price movement,
- Table A.33 represents the results obtained when the accuracy threshold is changed,
- Table A.34 illustrates the accuracy obtained when the direction of price movement is predicted.

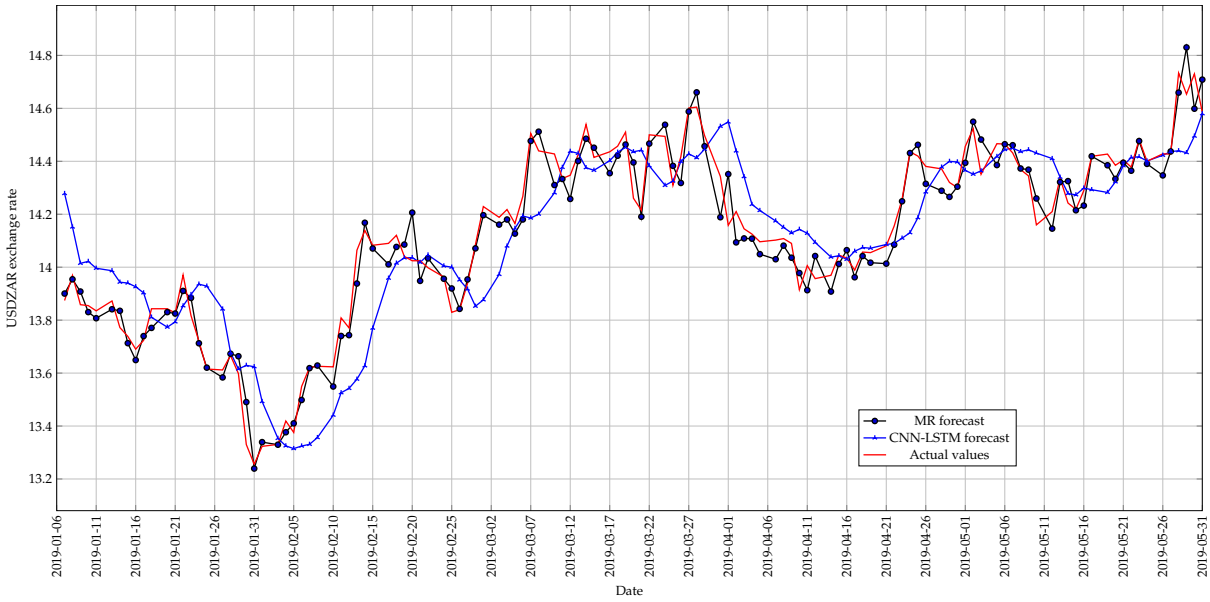
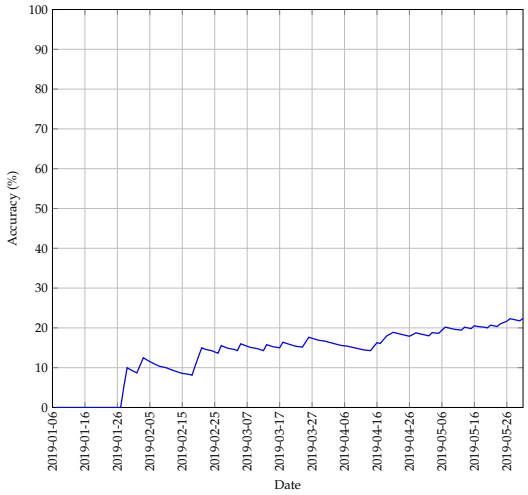
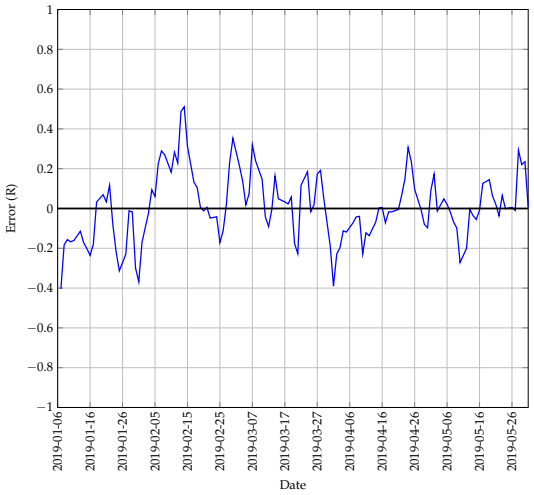


FIGURE A.81: *The best predictions made by the CNN-LSTM model for test 5 with 3 previous days as input using a walk-forward validation approach on the test set.*

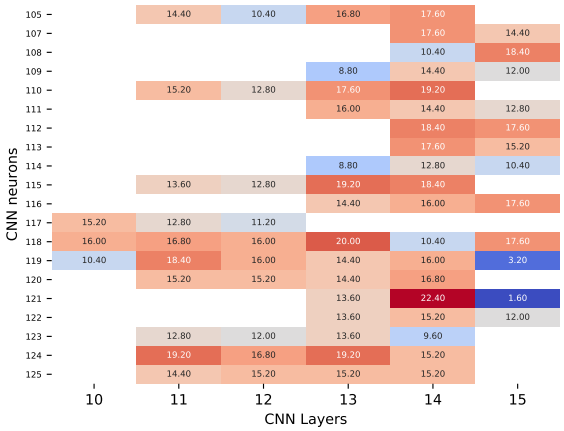


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

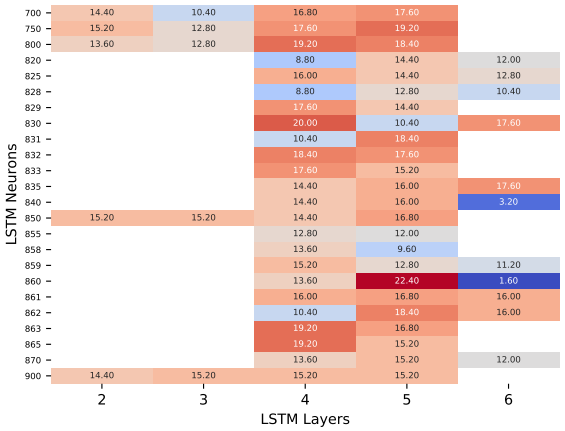


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.82: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 5.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 3 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 3 previous days as input.*

FIGURE A.83: *A graphical representation of the influence that different architectures have on prediction accuracy for test 5 of the CNN-LSTM model.*

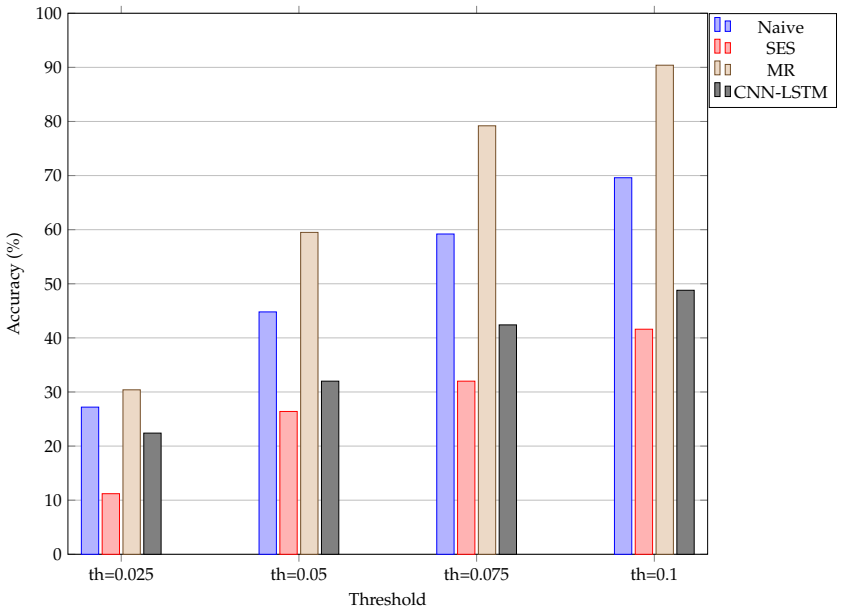


FIGURE A.84: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	22.4	32.0	42.4	48.8

TABLE A.33: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	51.613

TABLE A.34: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5.*

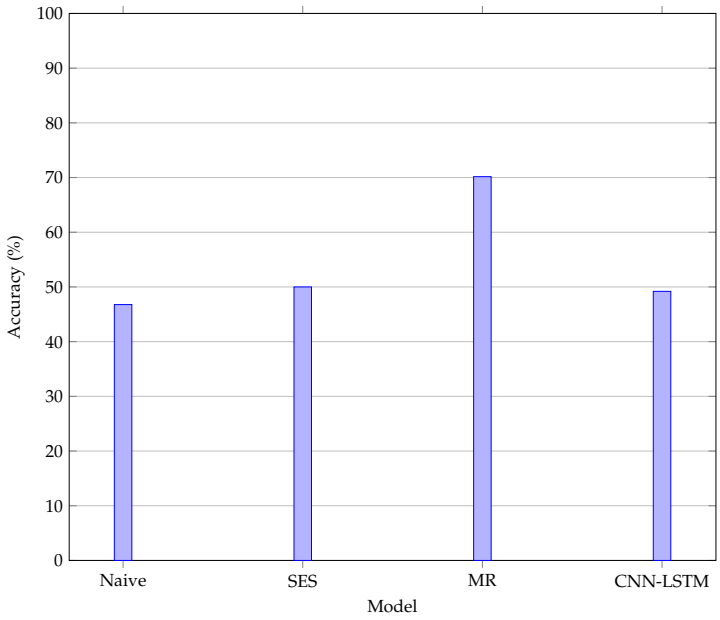


FIGURE A.85: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5.*

A.2.6 Test 6 results

The results obtained for test 6 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.86 shows the predictions made by the CNN-LSTM model,
- Figure A.87(a) and Figure A.87(b) show the change in accuracy and error as predictions are made,
- Figure A.88(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.88(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.89 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.90 shows the accuracy obtained when predicting the direction of price movement,
- Table A.35 represents the results obtained when the accuracy threshold is changed,
- Table A.36 illustrates the accuracy obtained when the direction of price movement is predicted.

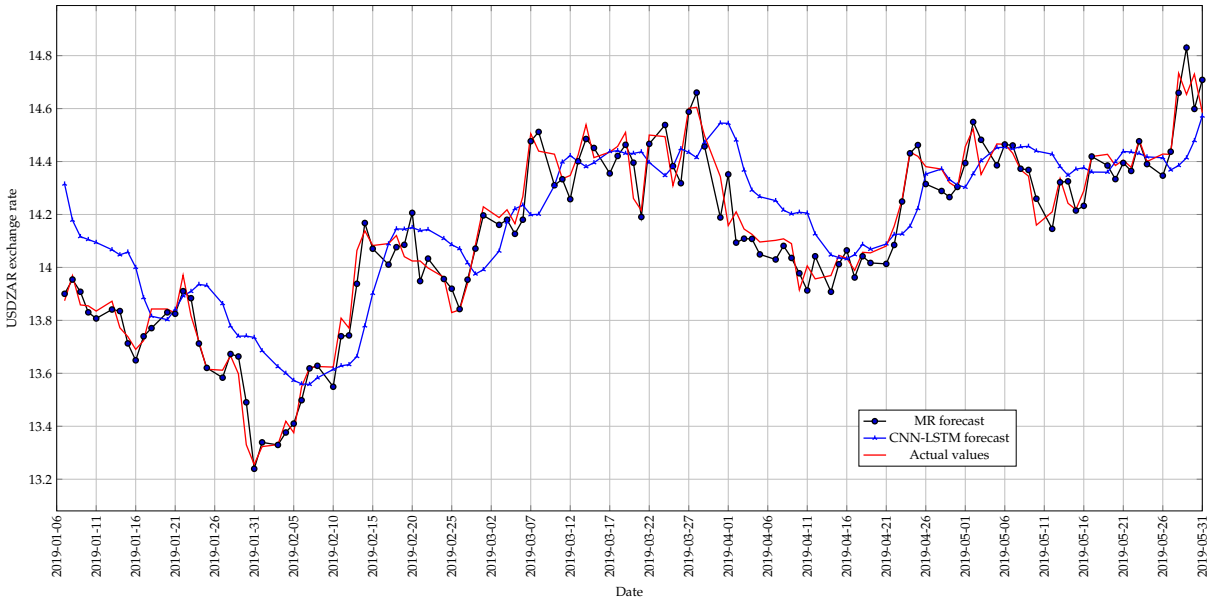
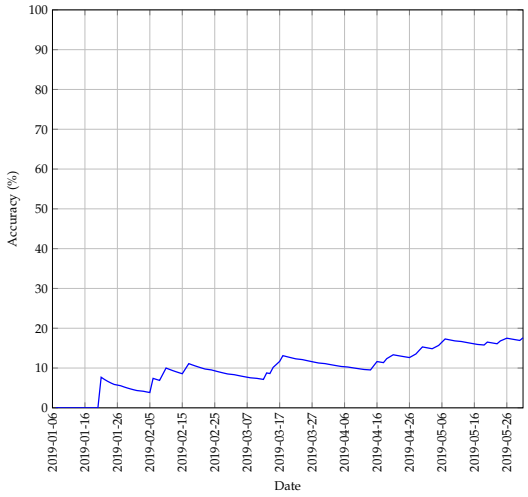
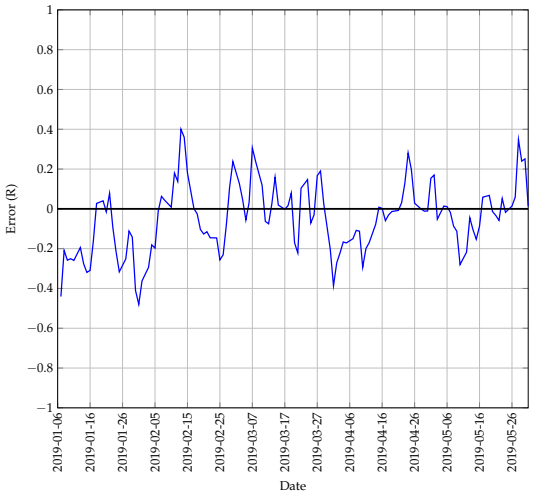


FIGURE A.86: *The best predictions made by the CNN-LSTM model for test 6 with 2 previous days as input using a walk-forward validation approach on the test set.*

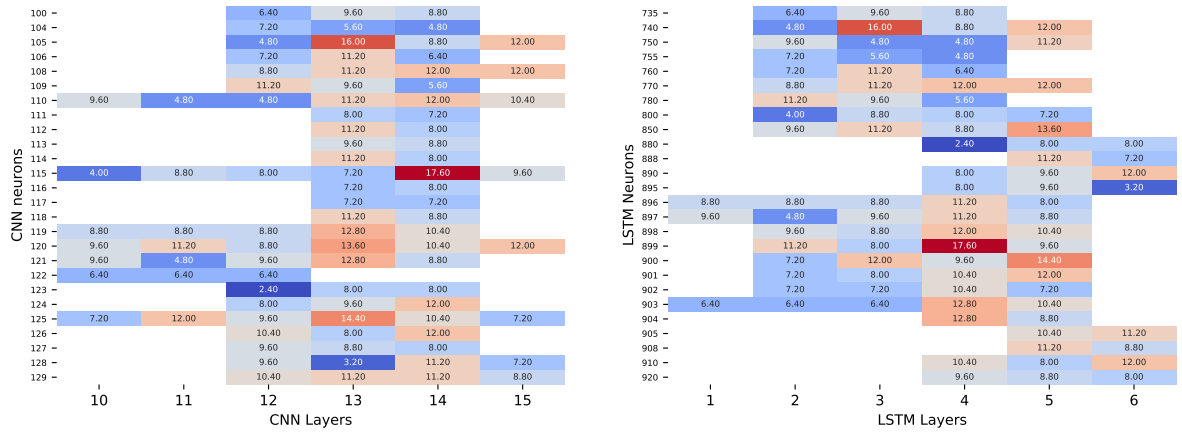


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.87: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 6.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 2 previous days as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 2 previous days as input.*

FIGURE A.88: *A graphical representation of the influence that different architectures have on prediction accuracy for test 6 of the CNN-LSTM model.*

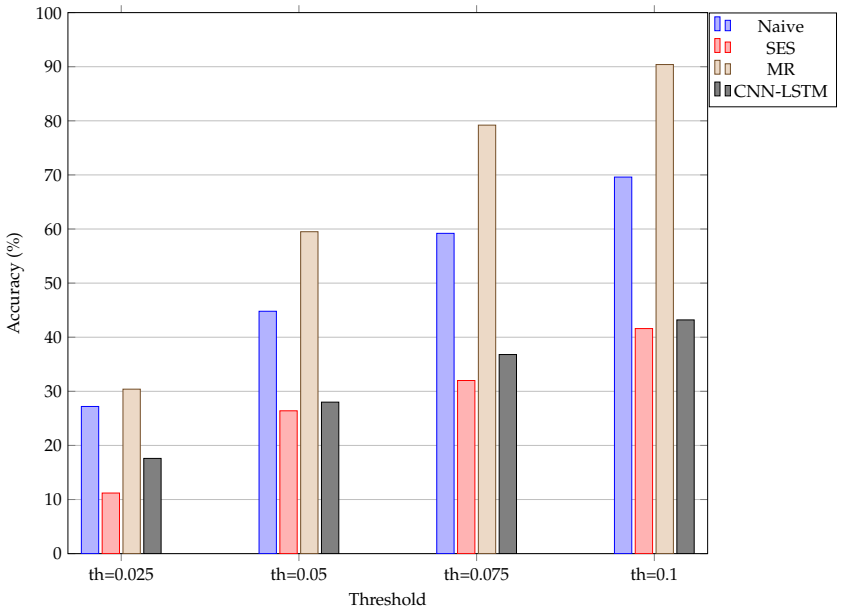


FIGURE A.89: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	17.6	28.0	36.8	43.2

TABLE A.35: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6.*

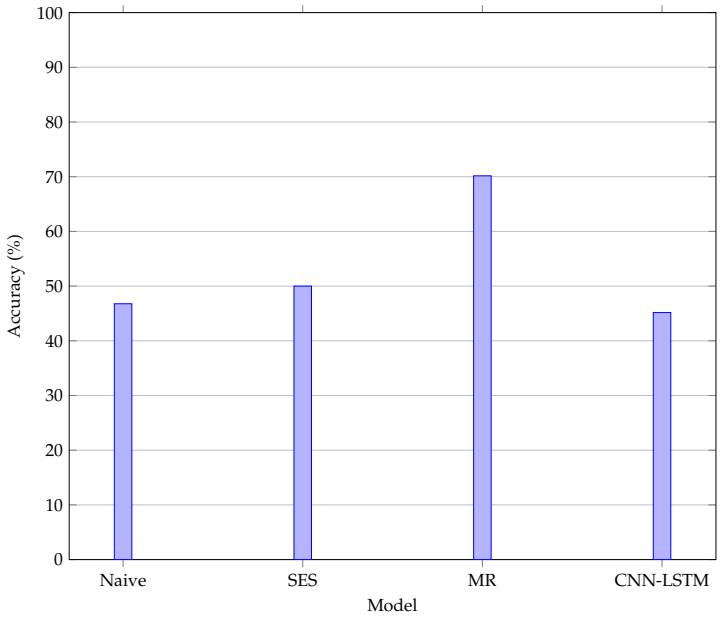


FIGURE A.90: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	48.387

TABLE A.36: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6.*

A.2.7 Test 7 results

The results obtained for test 7 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.91 shows the predictions made by the CNN-LSTM model,
- Figure A.92(a) and Figure A.92(b) show the change in accuracy and error as predictions are made,
- Figure A.93(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.93(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.94 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.95 shows the accuracy obtained when predicting the direction of price movement,
- Table A.37 represents the results obtained when the accuracy threshold is changed,
- Table A.38 illustrates the accuracy obtained when the direction of price movement is predicted.

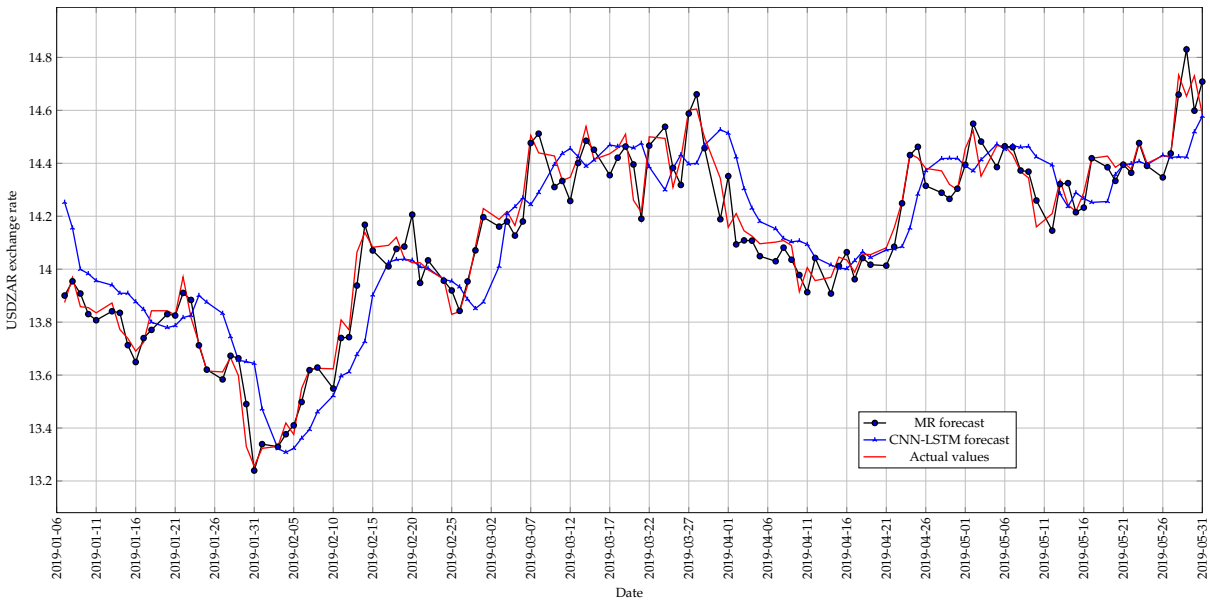
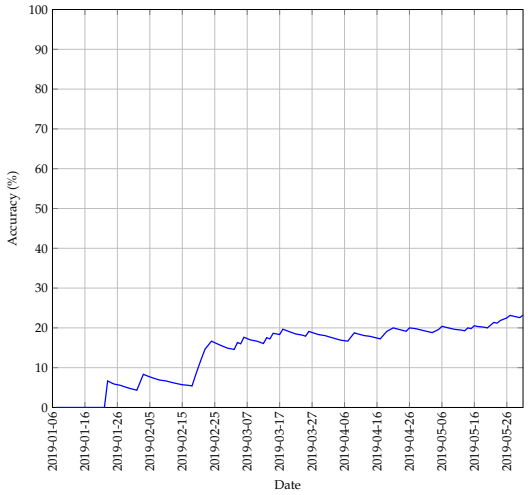
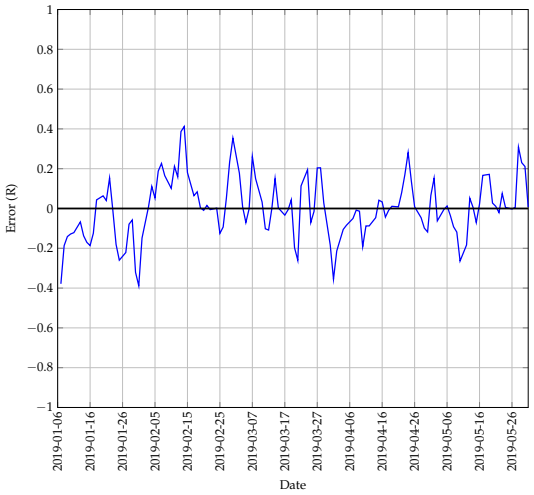


FIGURE A.91: *The best predictions made by the CNN-LSTM model for test 7 with 3 previous days as input using a walk-forward validation approach on the test set.*

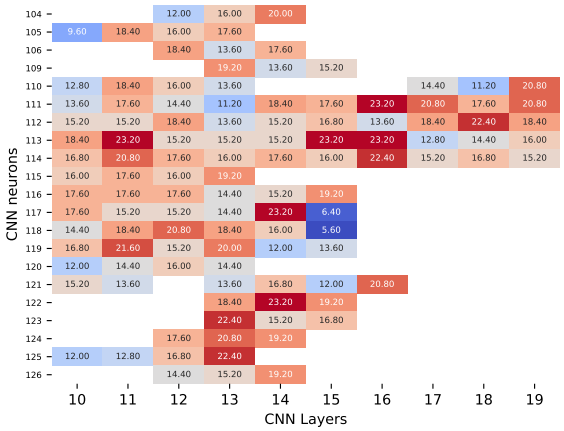


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

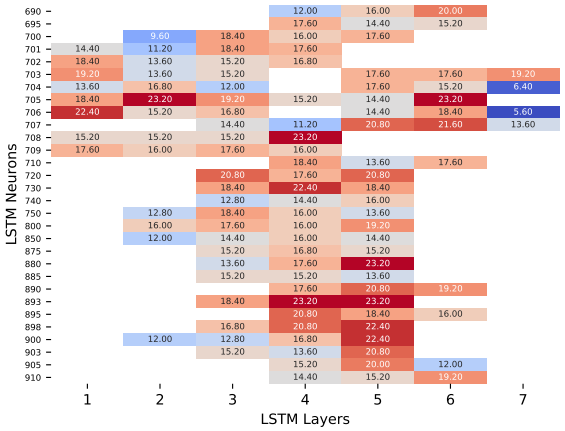


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.92: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 7.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 3 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 3 previous days as input.*

FIGURE A.93: *A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the CNN-LSTM model.*

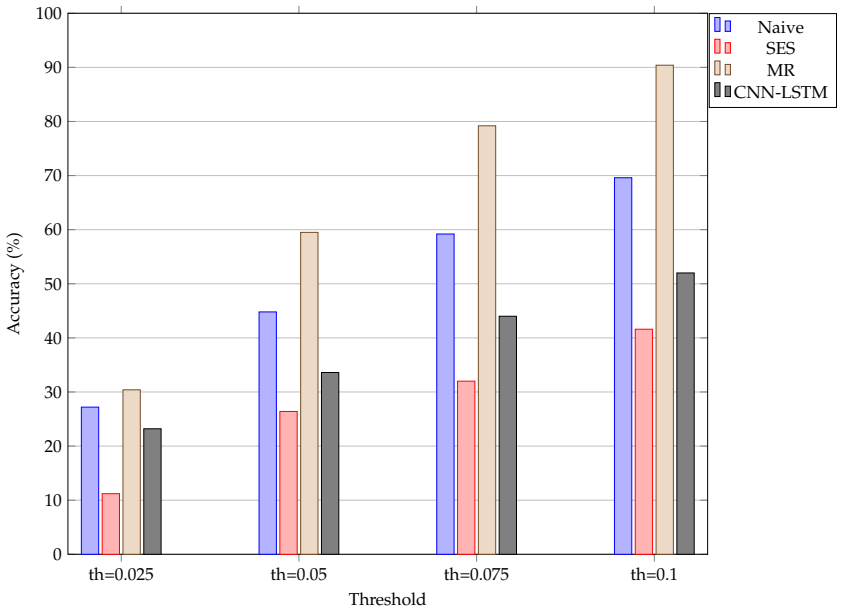


FIGURE A.94: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	23.2	33.6	44.0	52.0

TABLE A.37: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	51.613

TABLE A.38: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7.*

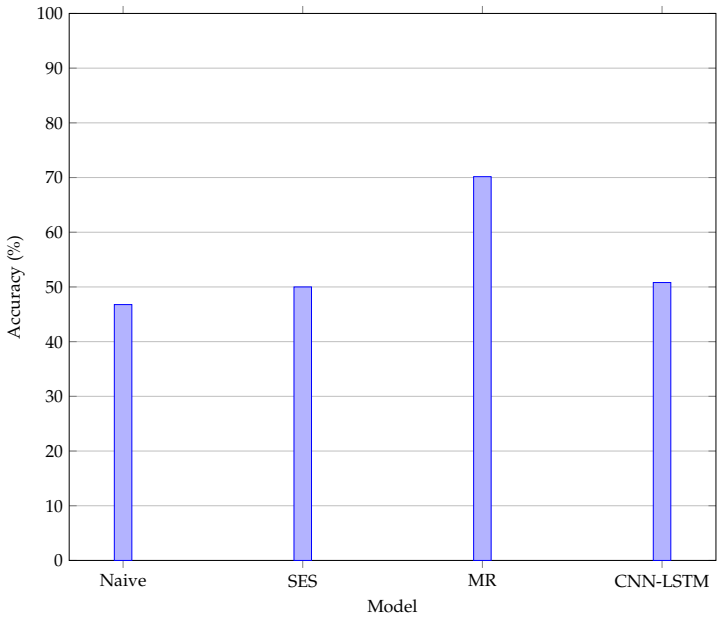


FIGURE A.95: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7.*

A.2.8 Test 8 results

The results obtained for test 8 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.96 shows the predictions made by the CNN-LSTM model,
- Figure A.97(a) and Figure A.97(b) show the change in accuracy and error as predictions are made,
- Figure A.98(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.98(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.99 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.100 shows the accuracy obtained when predicting the direction of price movement,
- Table A.39 represents the results obtained when the accuracy threshold is changed,
- Table A.40 illustrates the accuracy obtained when the direction of price movement is predicted.

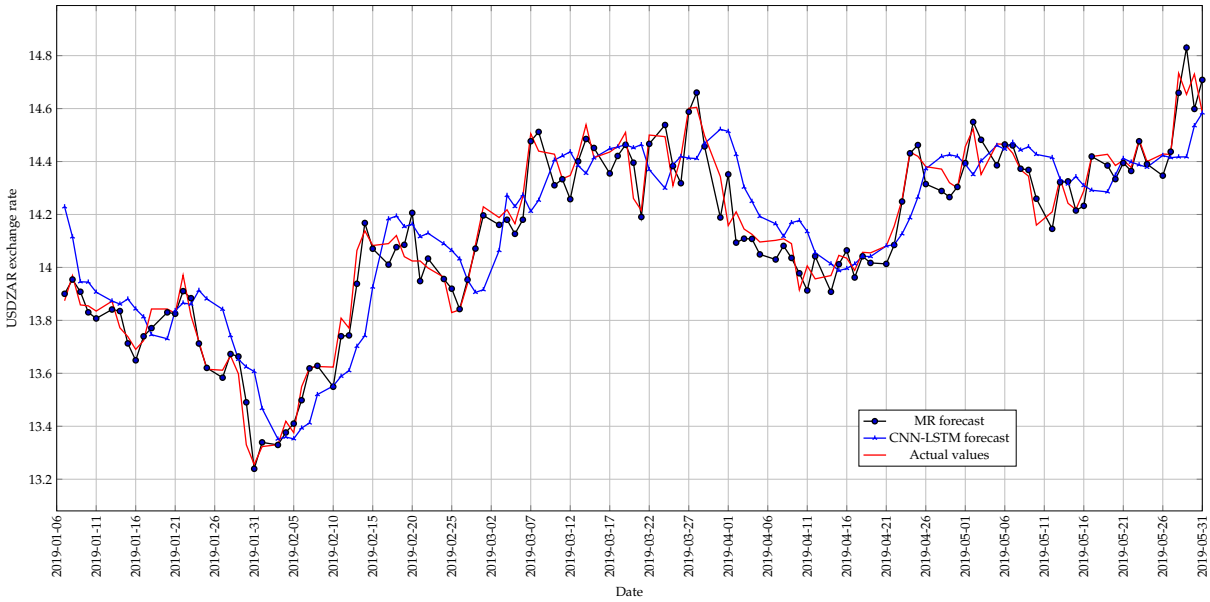
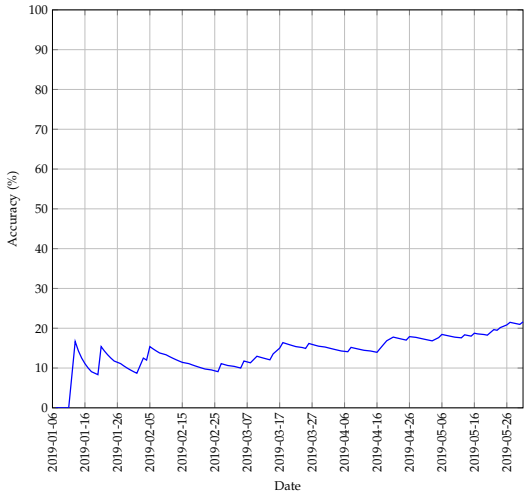
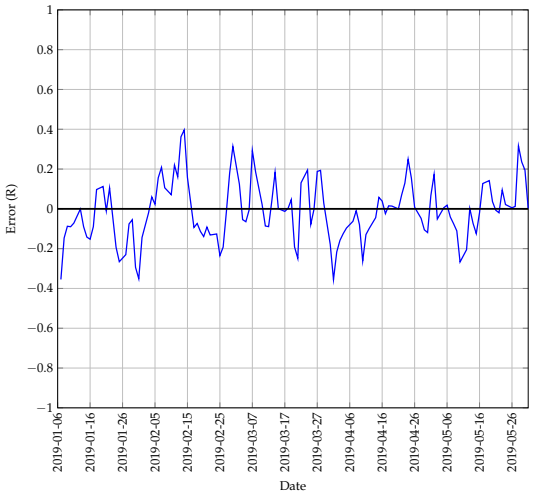


FIGURE A.96: *The best predictions made by the CNN-LSTM model for test 8 with 3 previous days as input using a walk-forward validation approach on the test set.*

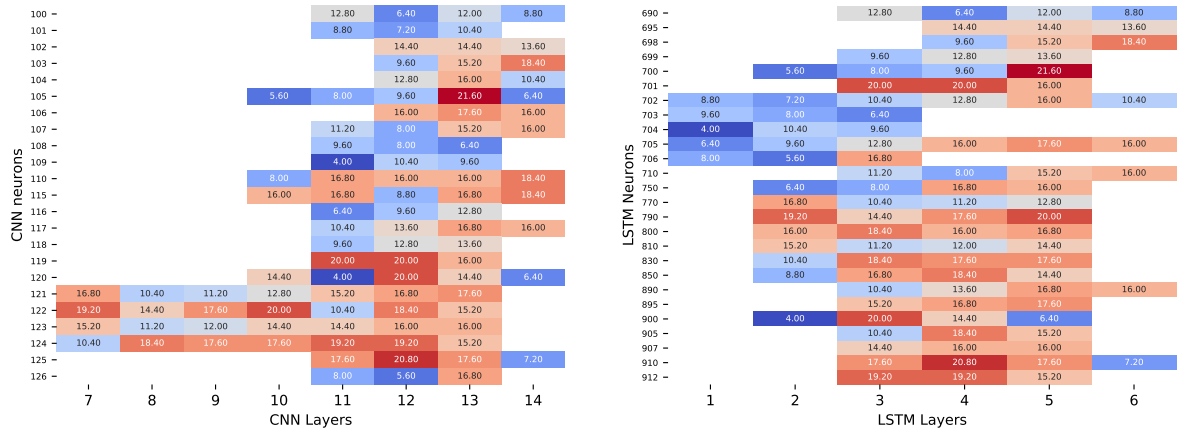


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.97: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 8.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 3 previous days as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 3 previous days as input.*

FIGURE A.98: *A graphical representation of the influence that different architectures have on prediction accuracy for test 8 of the CNN-LSTM model.*

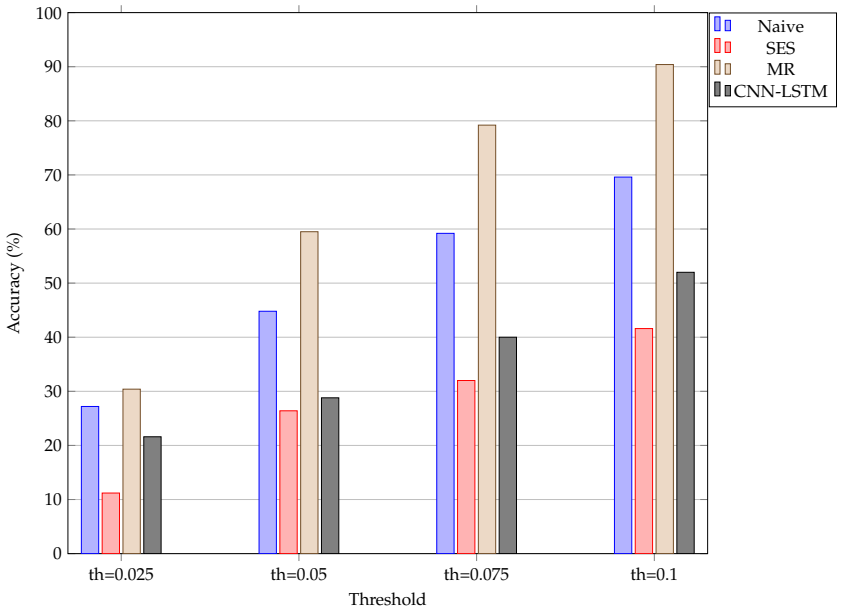


FIGURE A.99: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	21.6	28.8	40.0	52.0

TABLE A.39: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8.*

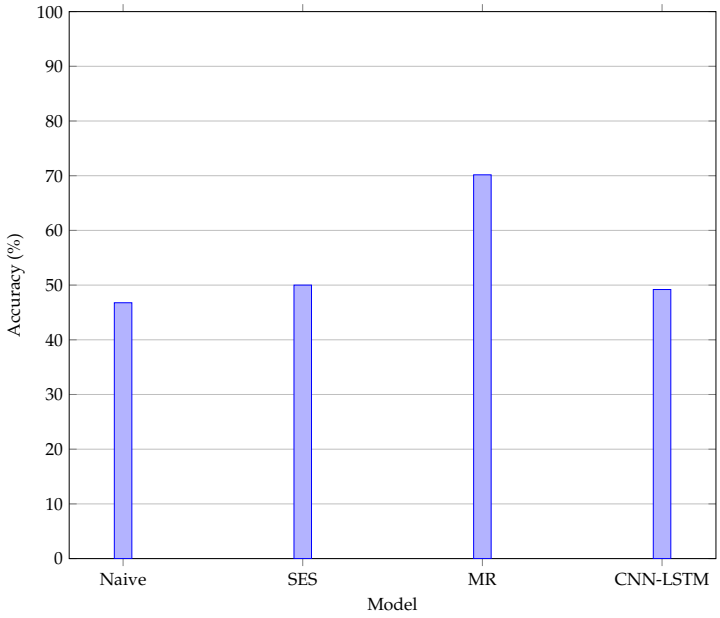


FIGURE A.100: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	50.806

TABLE A.40: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8.*

A.2.9 Test 9 results

The results obtained for test 9 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.101 shows the predictions made by the CNN-LSTM model,
- Figure A.102(a) and Figure A.102(b) show the change in accuracy and error as predictions are made,
- Figure A.103(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.103(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.104 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.105 shows the accuracy obtained when predicting the direction of price movement,
- Table A.41 represents the results obtained when the accuracy threshold is changed,
- Table A.42 illustrates the accuracy obtained when the direction of price movement is predicted.

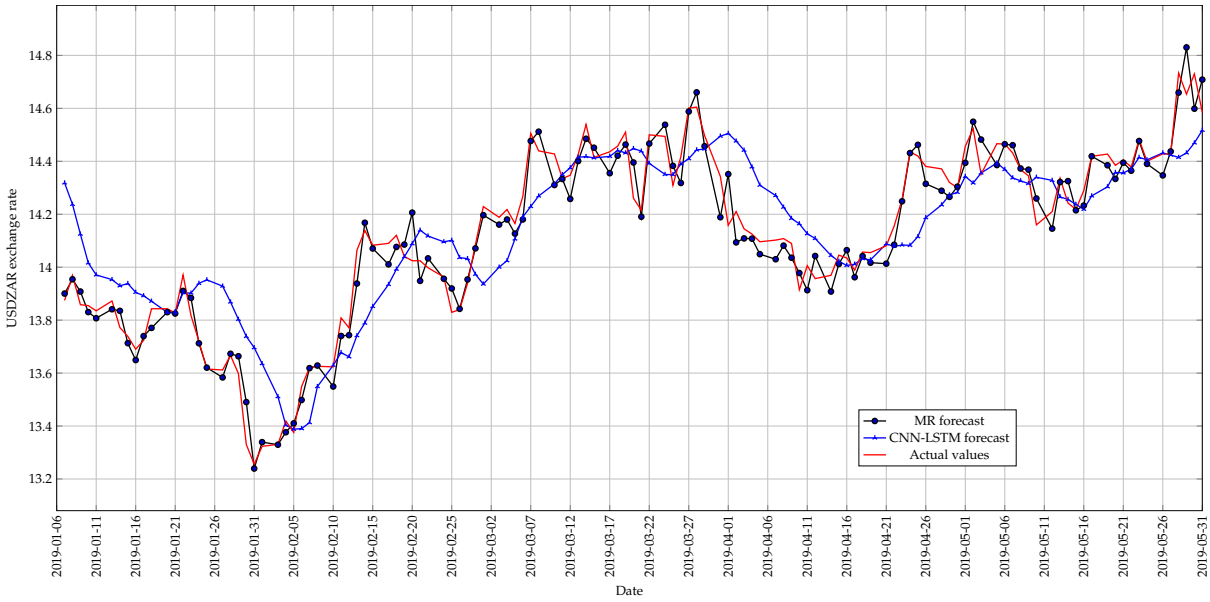
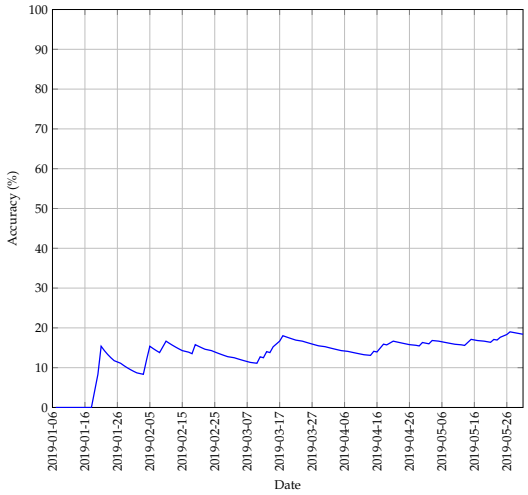
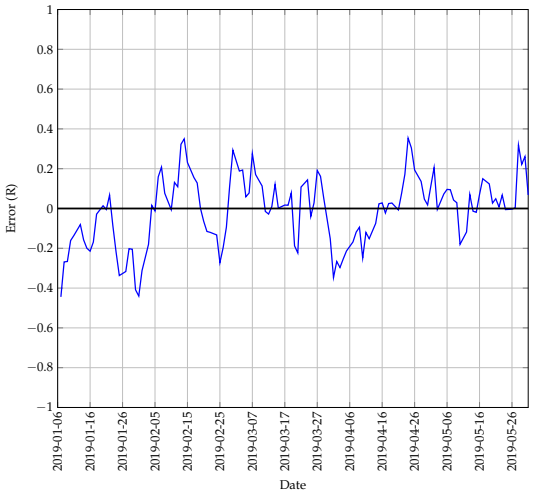


FIGURE A.101: *The best predictions made by the CNN-LSTM model for test 9 with 4 previous days as input using a walk-forward validation approach on the test set.*

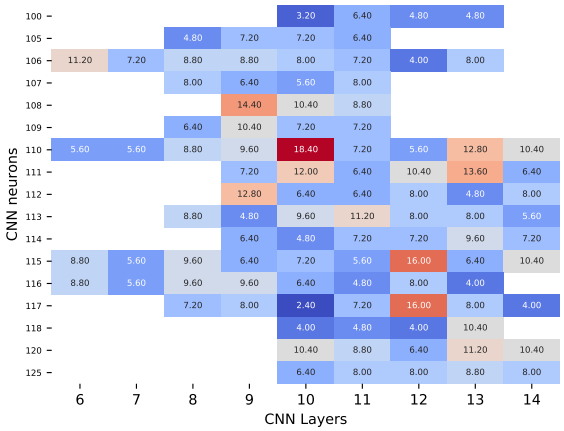


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

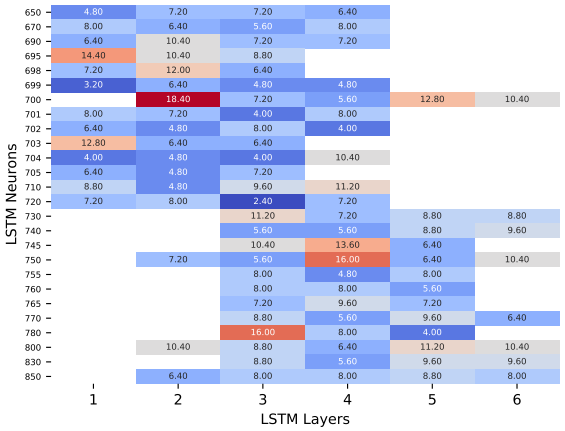


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.102: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 9.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 4 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 4 previous days as input.*

FIGURE A.103: *A graphical representation of the influence that different architectures have on prediction accuracy for test 9 of the CNN-LSTM model.*

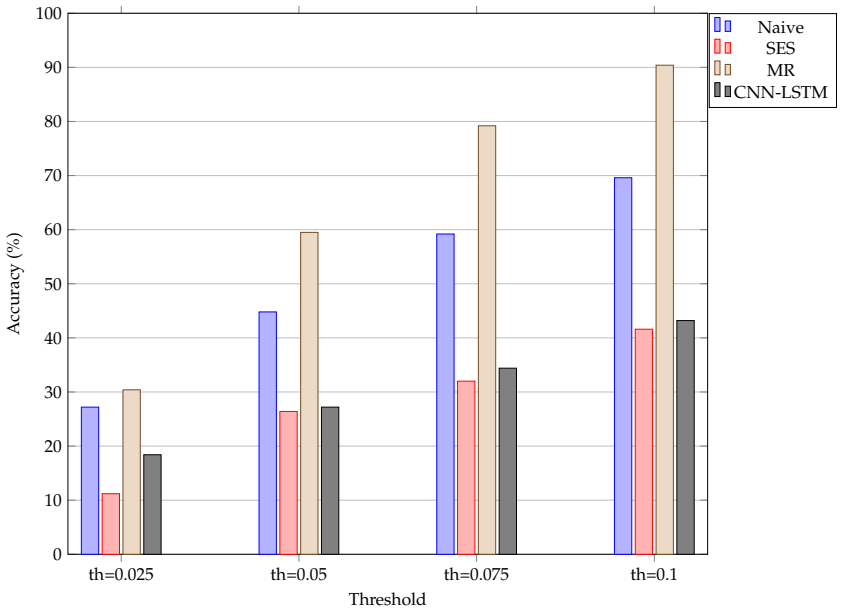


FIGURE A.104: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	18.4	27.2	34.4	43.2

TABLE A.41: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	50.0

TABLE A.42: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9.*

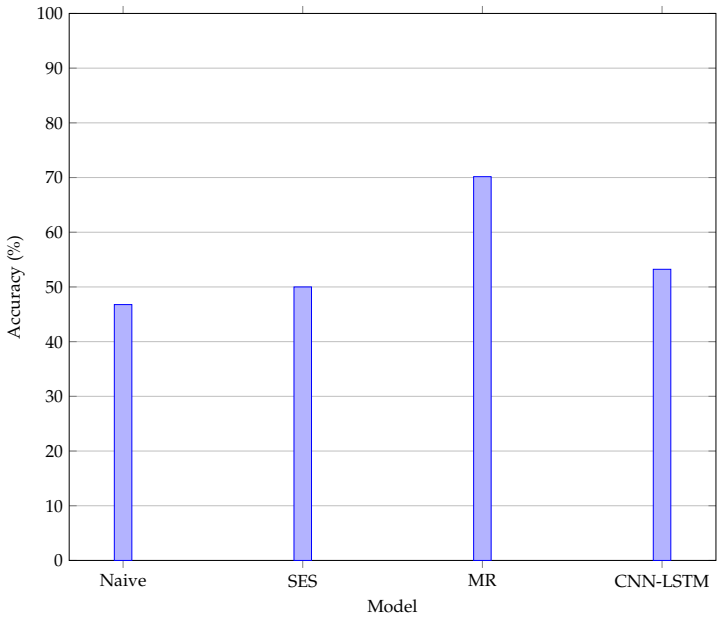


FIGURE A.105: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9.*

A.2.10 Test 10 results

The results obtained for test 10 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.106 shows the predictions made by the CNN-LSTM model,
- Figure A.107(a) and Figure A.107(b) show the change in accuracy and error as predictions are made,
- Figure A.108(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.108(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.109 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.110 shows the accuracy obtained when predicting the direction of price movement,
- Table A.43 represents the results obtained when the accuracy threshold is changed,
- Table A.44 illustrates the accuracy obtained when the direction of price movement is predicted.

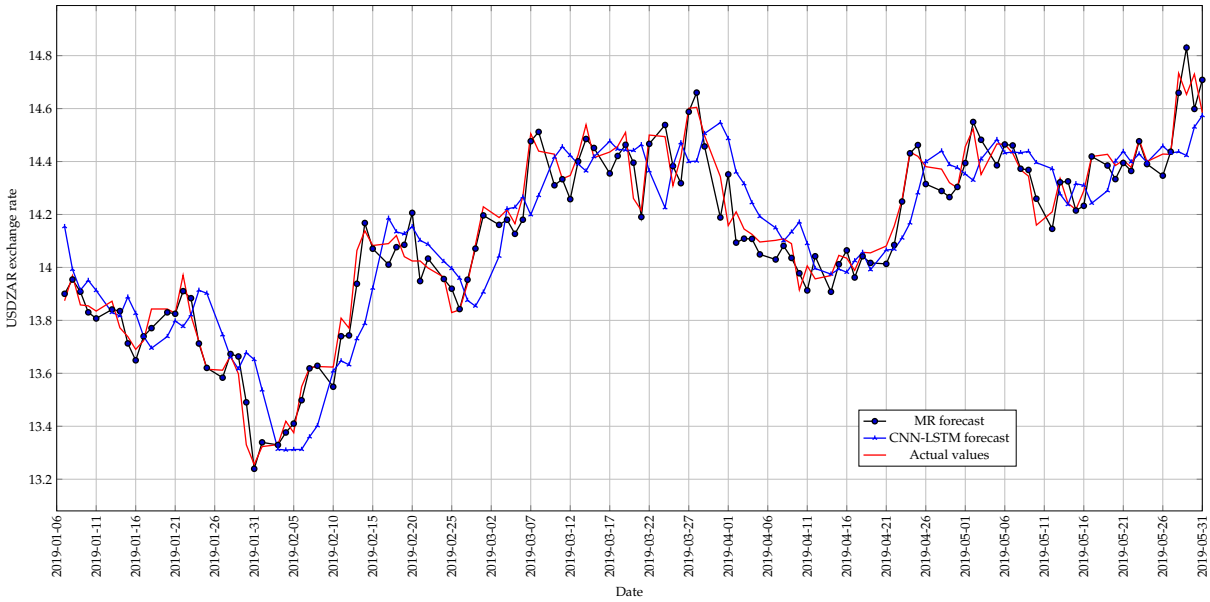
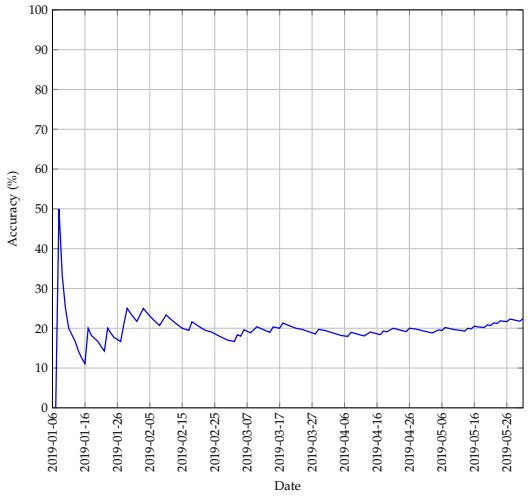
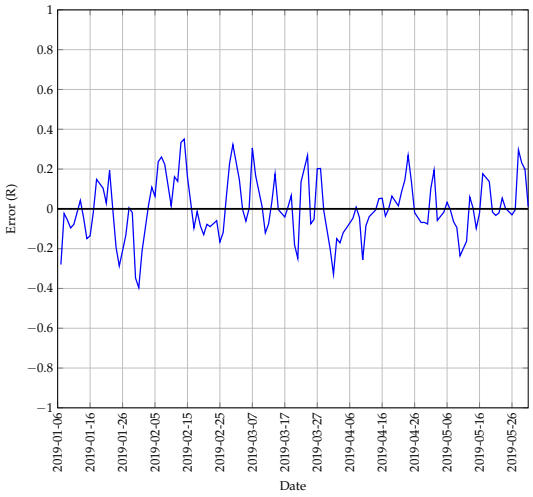


FIGURE A.106: *The best predictions made by the CNN-LSTM model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set.*

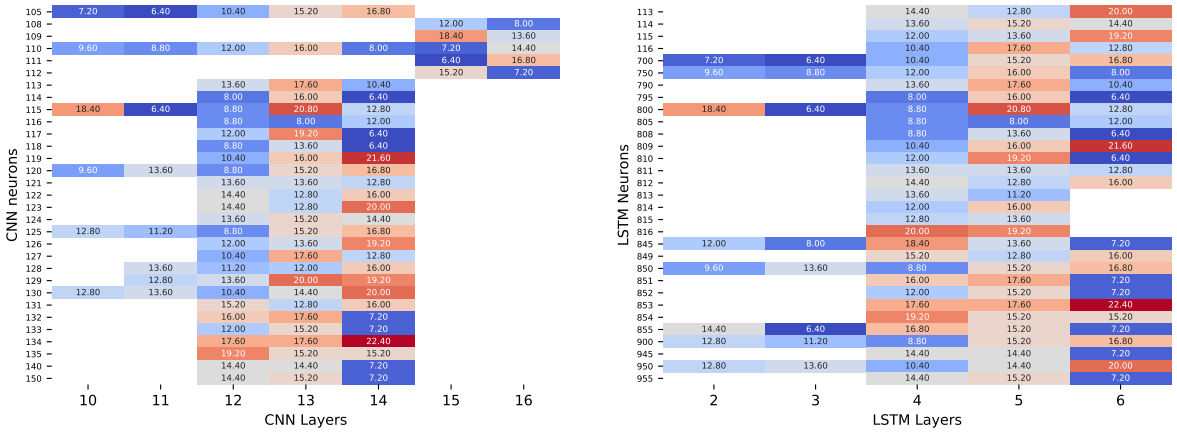


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.107: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 10.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 1 previous day as input.*

FIGURE A.108: *A graphical representation of the influence that different architectures have on prediction accuracy for test 10 of the CNN-LSTM model.*

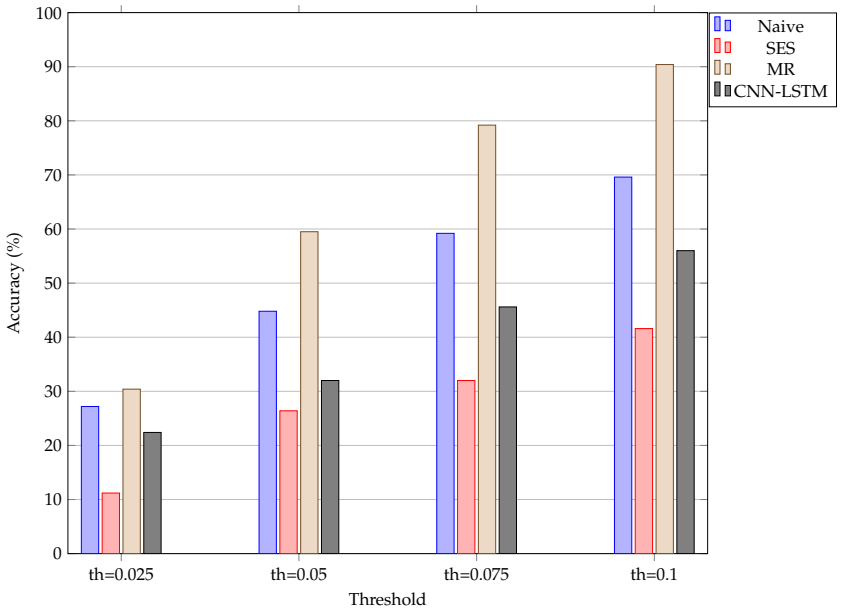


FIGURE A.109: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	22.4	32.0	45.6	56.0

TABLE A.43: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10.*

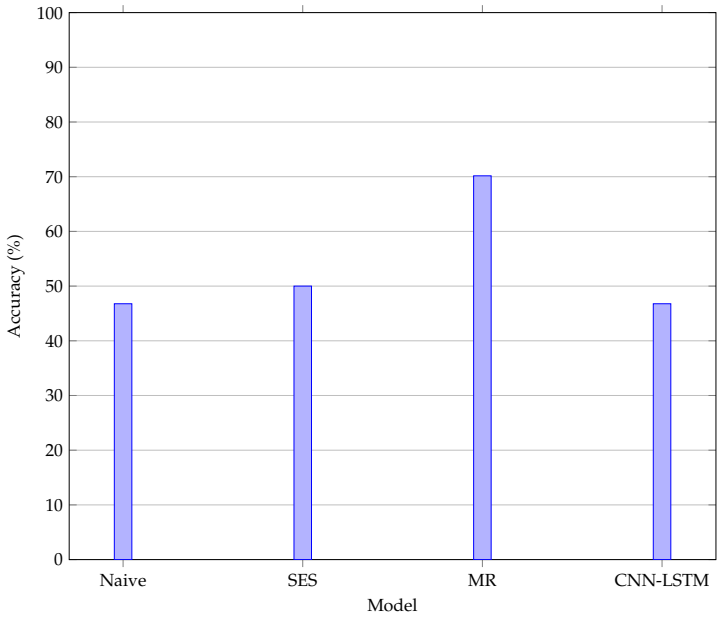


FIGURE A.110: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	42.742

TABLE A.44: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10.*

A.2.11 Test 11 results

The results obtained for test 11 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.111 shows the predictions made by the CNN-LSTM model,
- Figure A.112(a) and Figure A.112(b) show the change in accuracy and error as predictions are made,
- Figure A.113(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.113(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.114 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.115 shows the accuracy obtained when predicting the direction of price movement,
- Table A.45 represents the results obtained when the accuracy threshold is changed,
- Table A.46 illustrates the accuracy obtained when the direction of price movement is predicted.

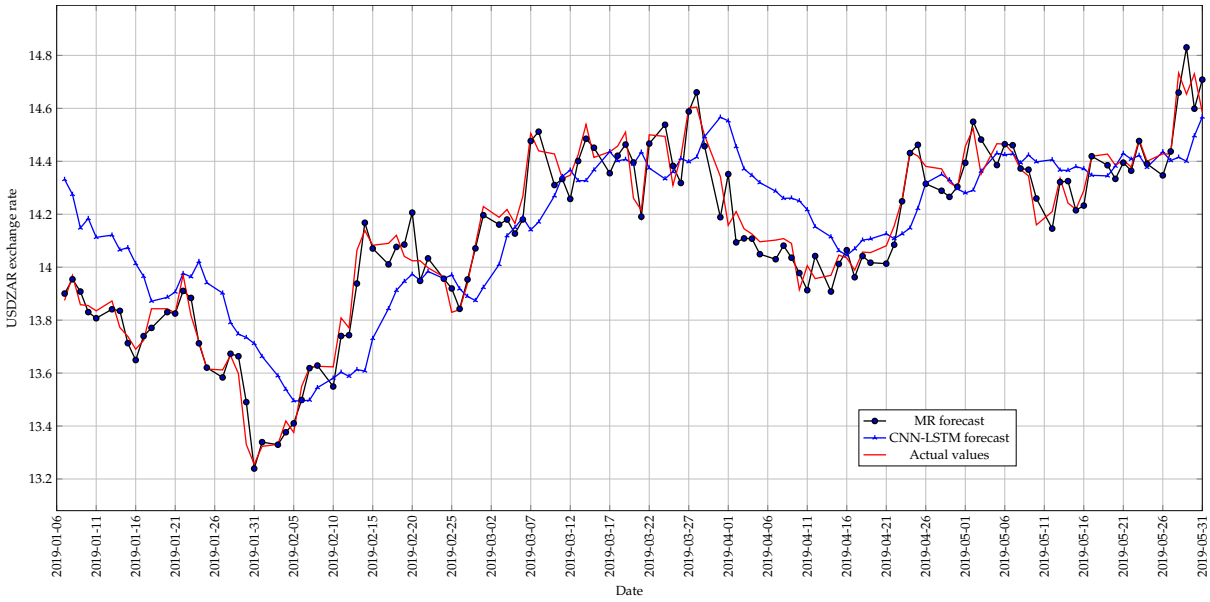
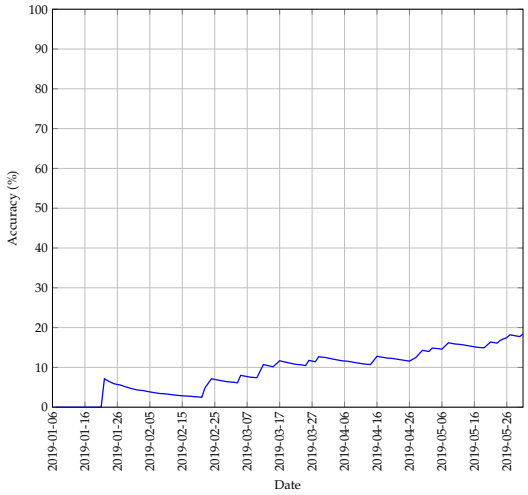
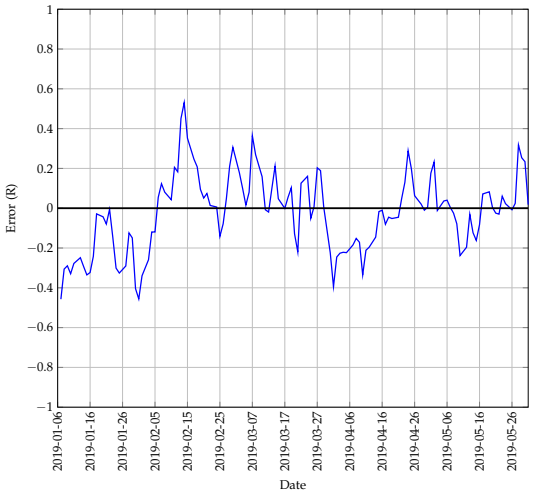


FIGURE A.111: *The best predictions made by the CNN-LSTM model for test 11 with 3 previous days as input using a walk-forward validation approach on the test set.*

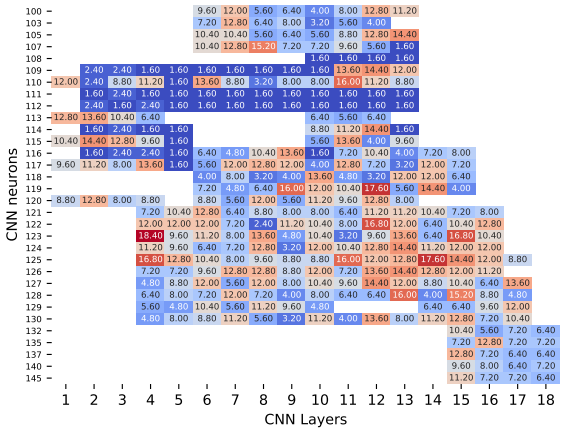


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*

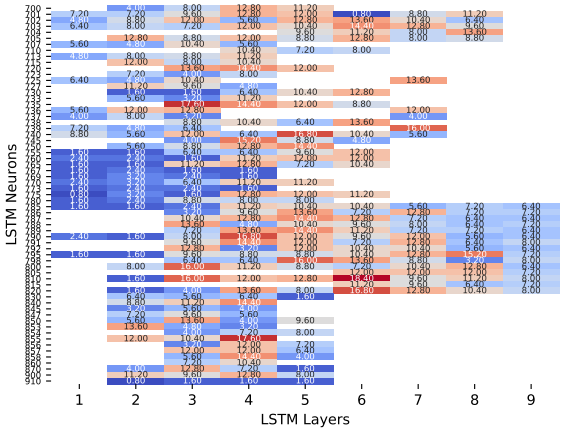


(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.112: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 11.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 3 previous days as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 3 previous days as input.*

FIGURE A.113: *A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the CNN-LSTM model.*

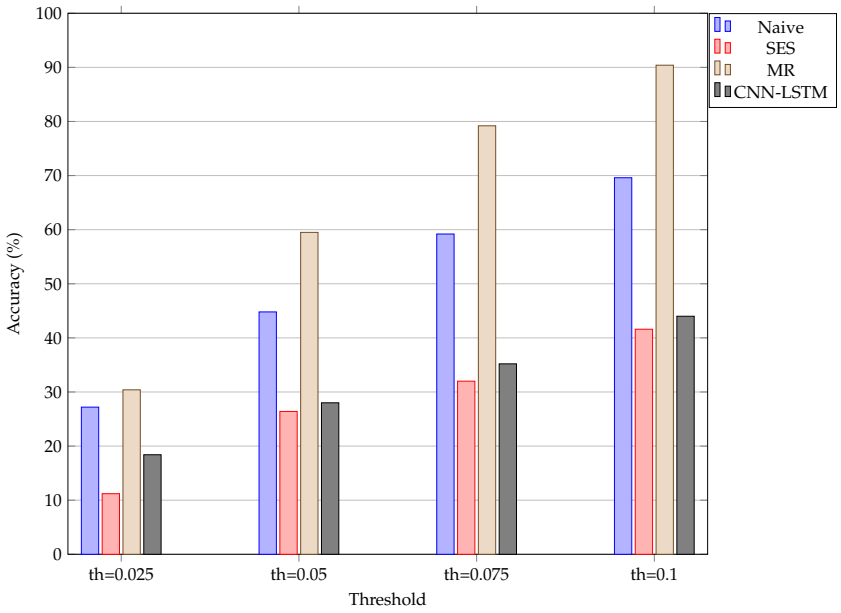


FIGURE A.114: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	18.4	28.0	35.2	44.0

TABLE A.45: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	56.452

TABLE A.46: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11.*

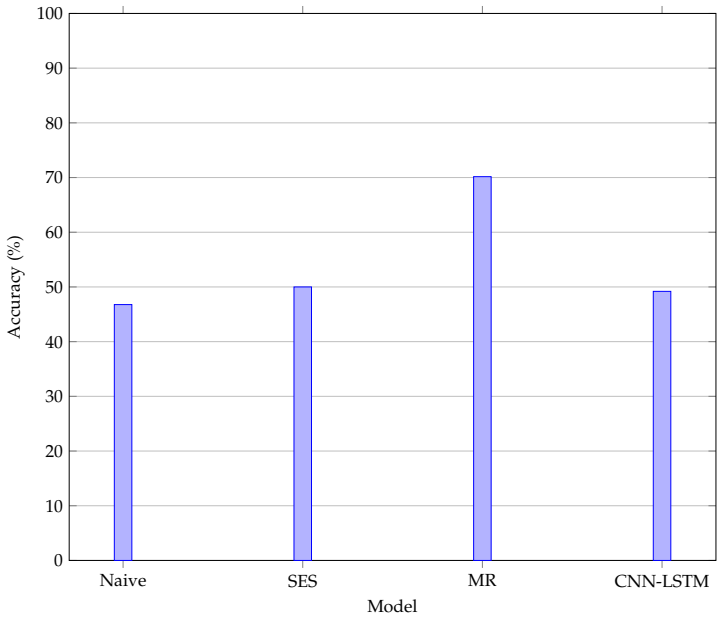


FIGURE A.115: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11.*

A.2.12 Test 12 results

The results obtained for test 12 of the CNN-LSTM model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.116 shows the predictions made by the CNN-LSTM model,
- Figure A.117(a) and Figure A.117(b) show the change in accuracy and error as predictions are made,
- Figure A.118(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.118(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.119 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.120 shows the accuracy obtained when predicting the direction of price movement,
- Table A.47 represents the results obtained when the accuracy threshold is changed,
- Table A.48 illustrates the accuracy obtained when the direction of price movement is predicted.

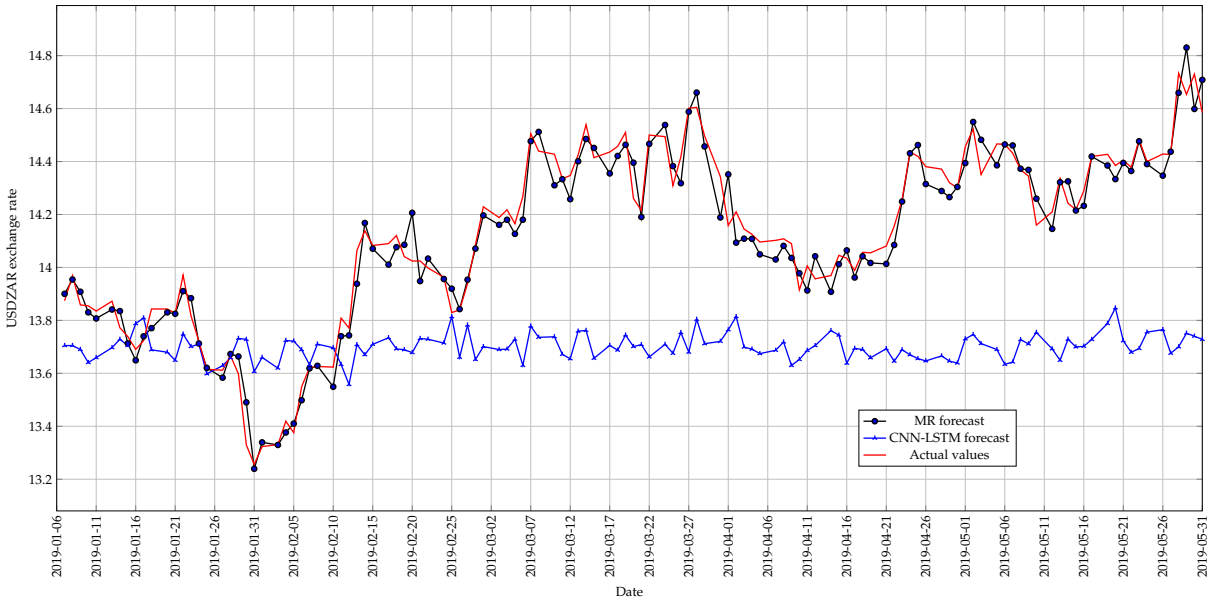
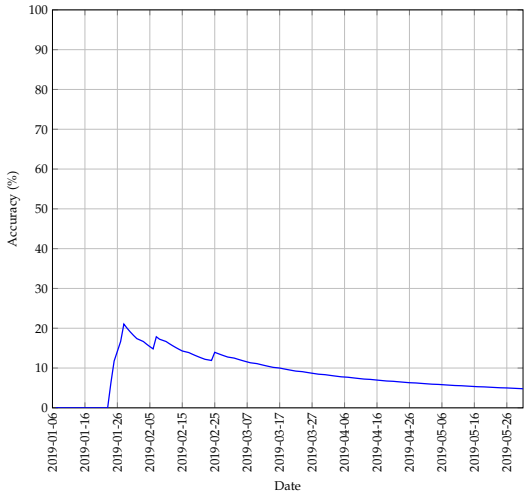
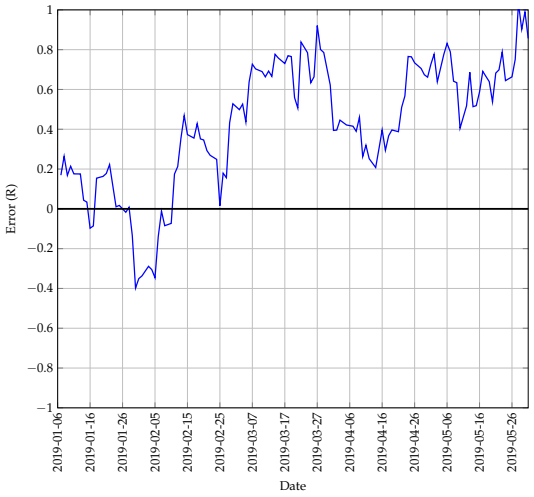


FIGURE A.116: *The best predictions made by the CNN-LSTM model for test 12 with 5 previous days as input using a walk-forward validation approach on the test set.*

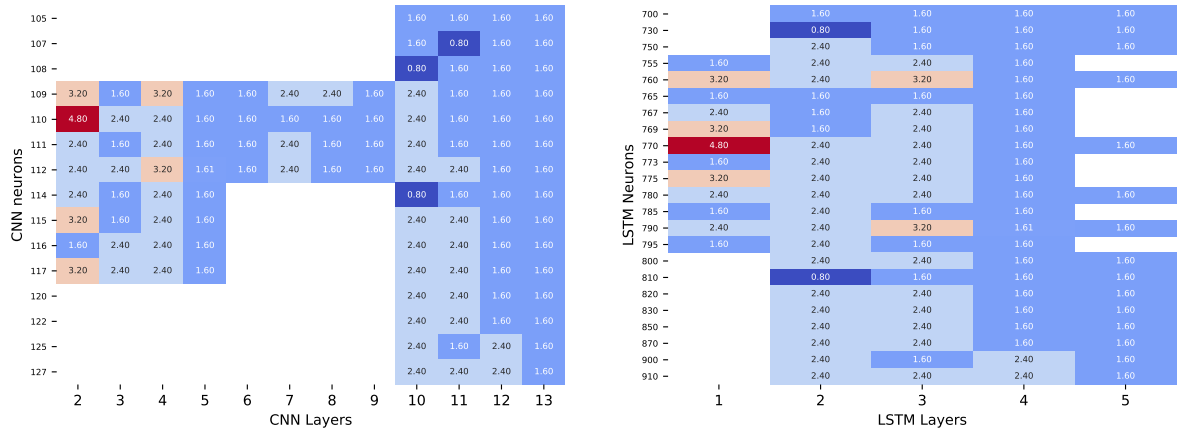


(a) *The change in accuracy of the CNN-LSTM model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the CNN-LSTM model whilst using a walk-forward validation approach on the test set.*

FIGURE A.117: *A graphical representation of how the accuracy and error of the CNN-LSTM model changed whilst making predictions on the test set for test 12.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the CNN-LSTM model with 5 previous days as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the CNN-LSTM model with 5 previous days as input.*

FIGURE A.118: *A graphical representation of the influence that different architectures have on prediction accuracy for test 12 of the CNN-LSTM model.*

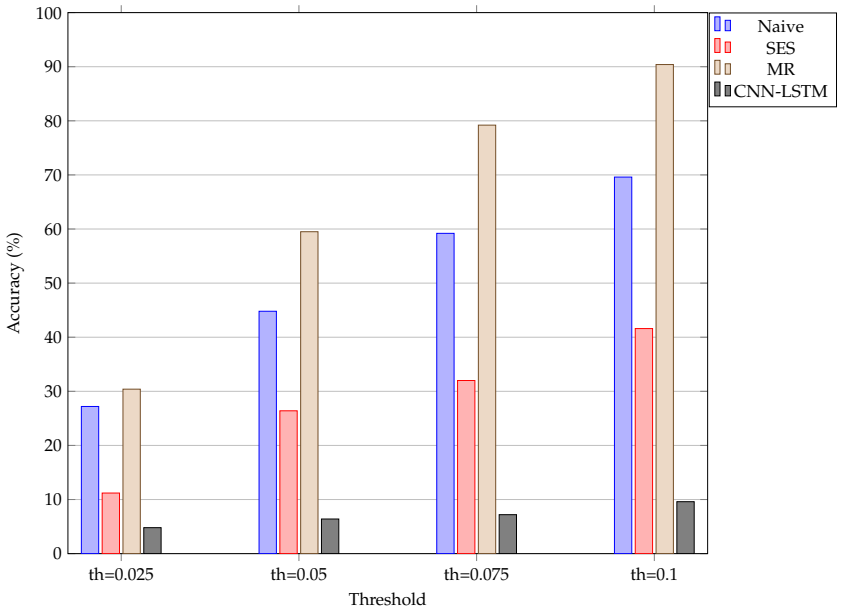


FIGURE A.119: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
CNN-LSTM	4.8	6.4	7.2	9.6

TABLE A.47: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12.*

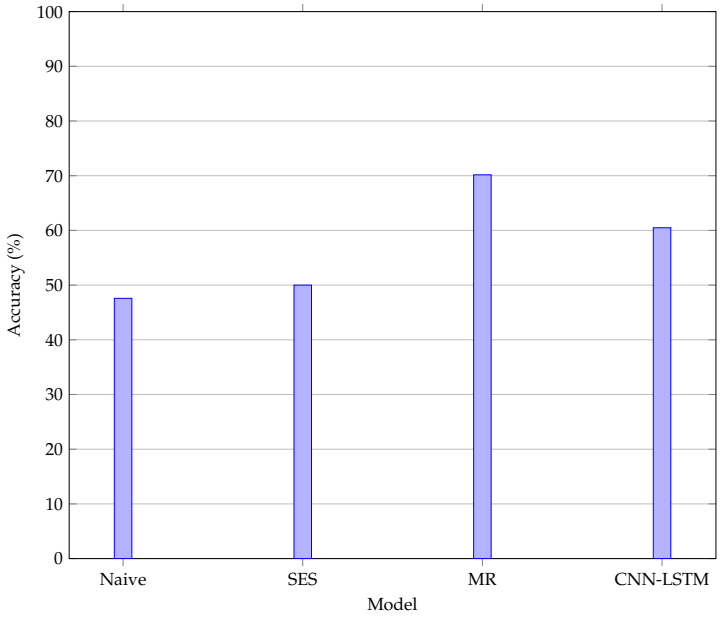


FIGURE A.120: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
CNN-LSTM	55.645

TABLE A.48: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12.*

A.3 Multi-head model results

This section contains the results obtained by the Multi-head model.

A.3.1 Test 1 results

The results obtained for test 1 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.121 shows the predictions made by the Multi-head model,
- Figure A.122(a) and Figure A.122(b) show the change in accuracy and error as predictions are made,
- Figure A.123(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.123(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.124 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.125 shows the accuracy obtained when predicting the direction of price movement,
- Table A.49 represents the results obtained when the accuracy threshold is changed,
- Table A.50 illustrates the accuracy obtained when the direction of price movement is predicted.

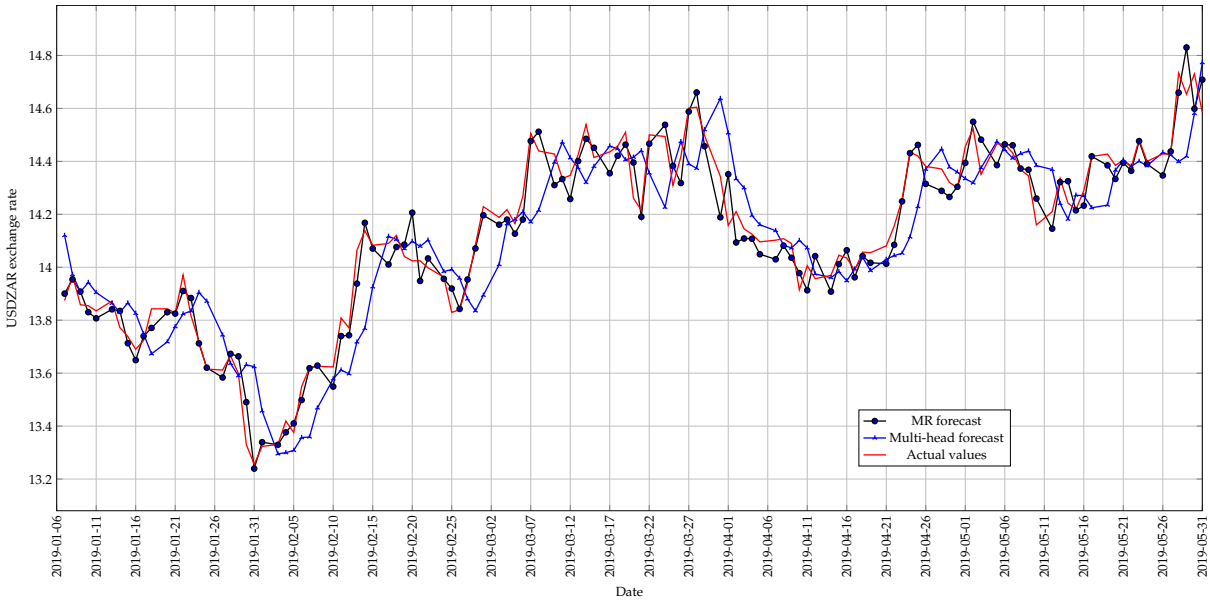
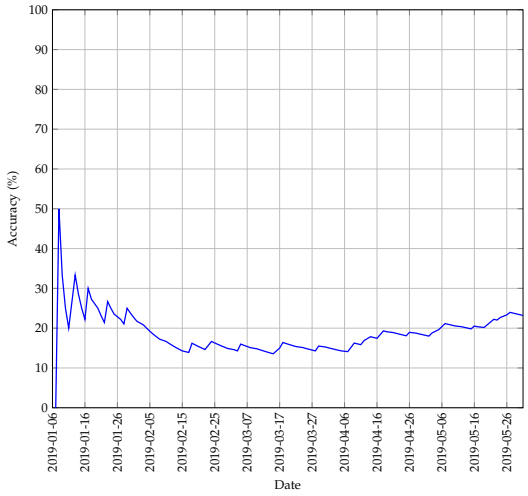
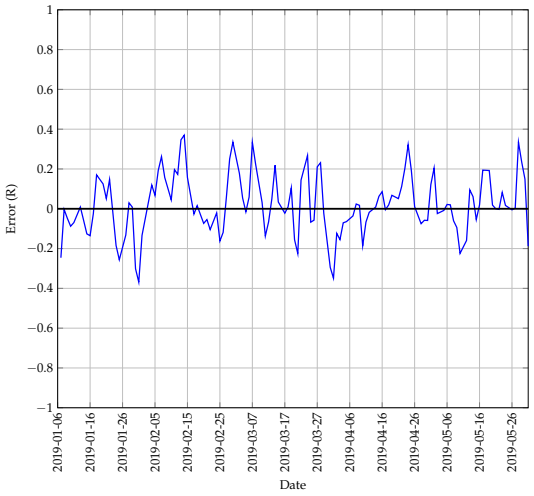


FIGURE A.121: *The best predictions made by the Multi-head model for test 1 with 1 previous day as input using a walk-forward validation approach on the test set.*

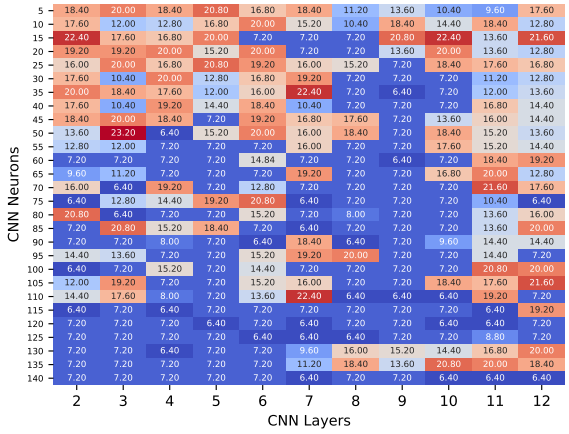


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

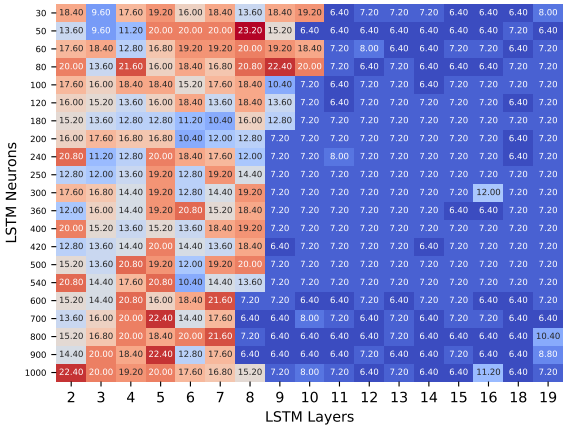


(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.122: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 1.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.123: *A graphical representation of the influence that different architectures have on prediction accuracy for test 1 of the Multi-head model.*

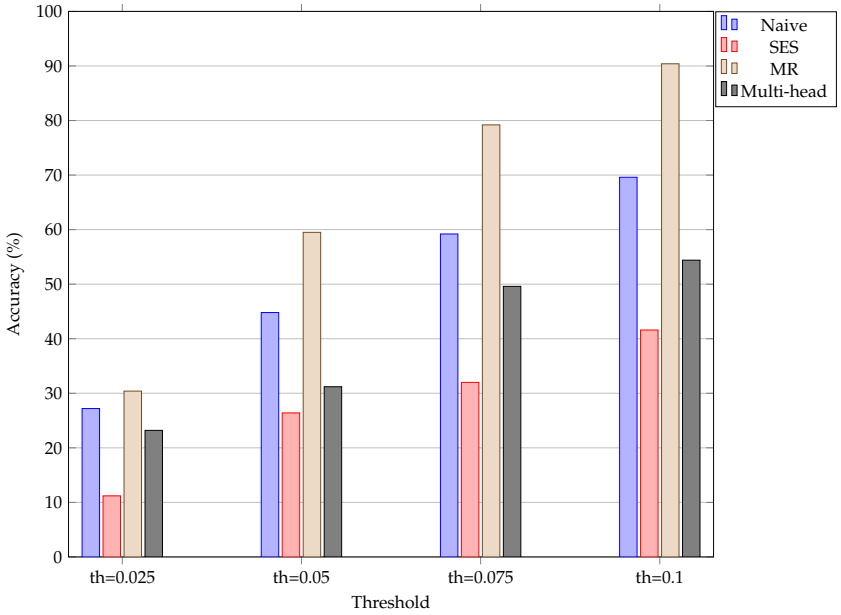


FIGURE A.124: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 1. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	23.2	31.2	49.6	54.4

TABLE A.49: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 1.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.50: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 1.*

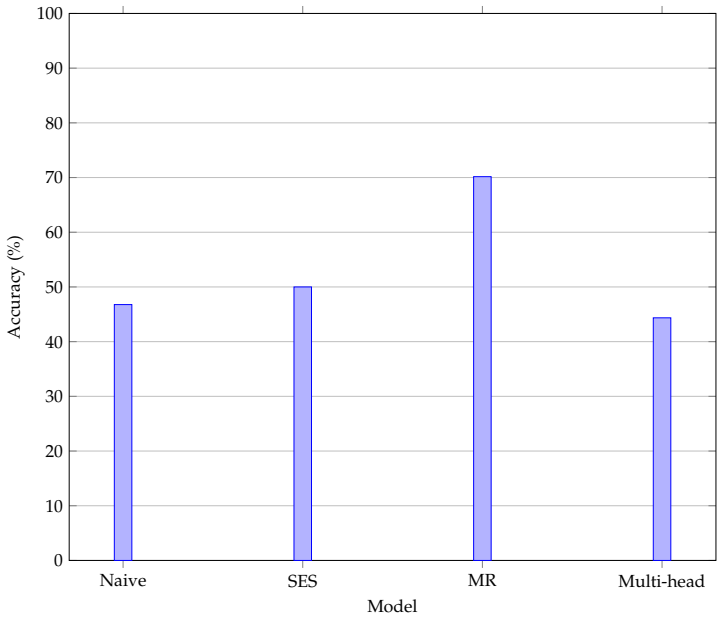


FIGURE A.125: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 1.*

A.3.2 Test 2 results

The results obtained for test 2 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.126 shows the predictions made by the Multi-head model,
- Figure A.127(a) and Figure A.127(b) show the change in accuracy and error as predictions are made,
- Figure A.128(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.128(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.129 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.130 shows the accuracy obtained when predicting the direction of price movement,
- Table A.51 represents the results obtained when the accuracy threshold is changed,
- Table A.52 illustrates the accuracy obtained when the direction of price movement is predicted.

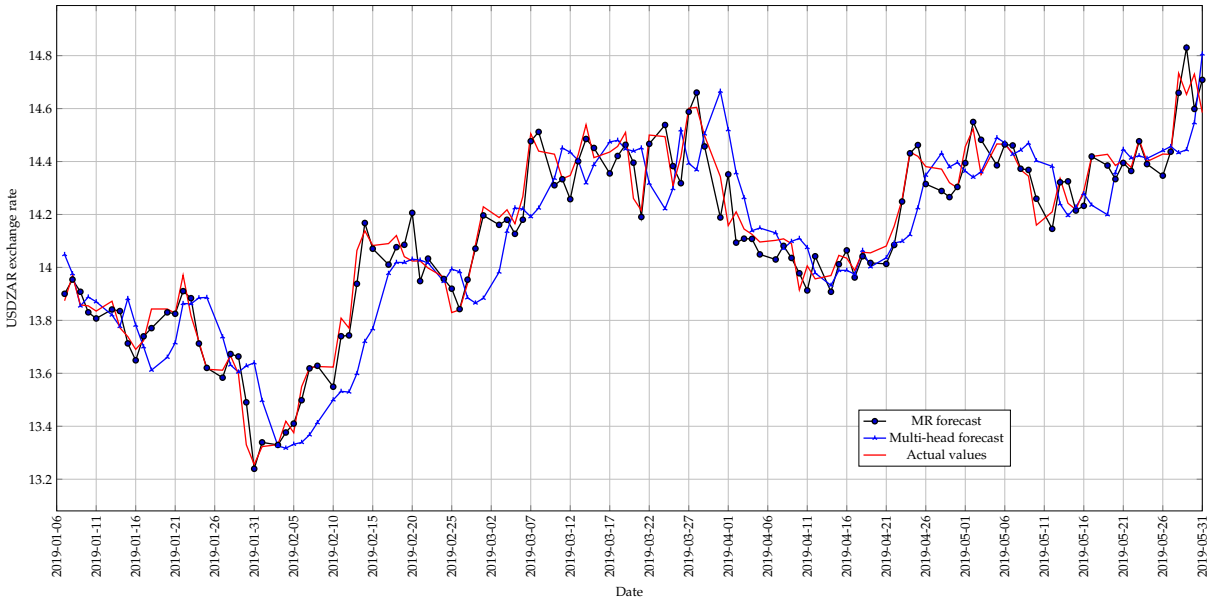
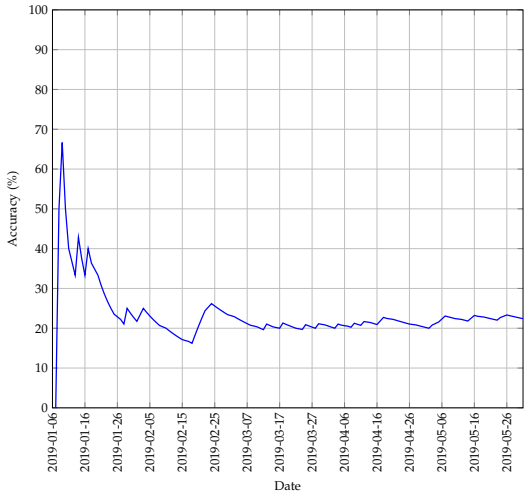
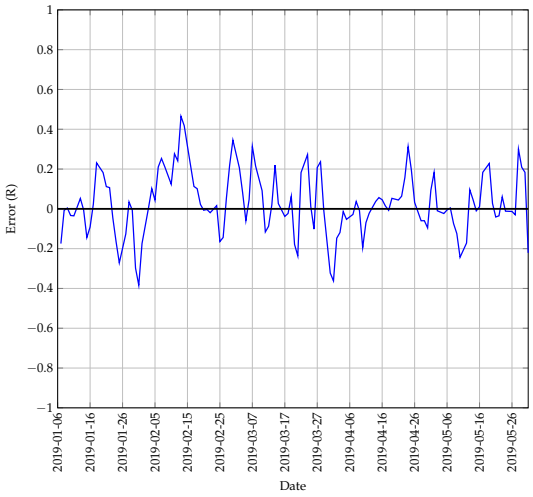


FIGURE A.126: *The best predictions made by the Multi-head model for test 2 with 1 previous day as input using a walk-forward validation approach on the test set.*

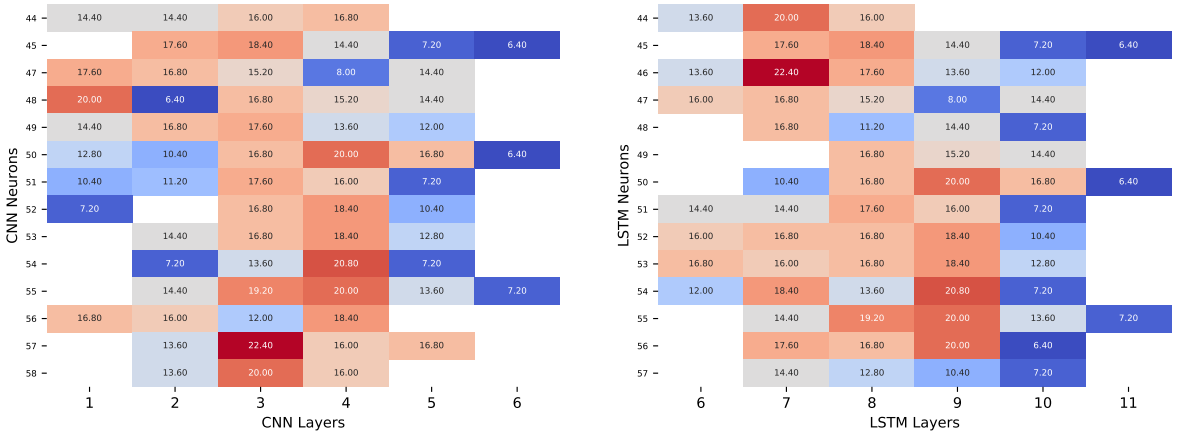


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.127: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 2.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.128: *A graphical representation of the influence that different architectures have on prediction accuracy for test 2 of the Multi-head model.*

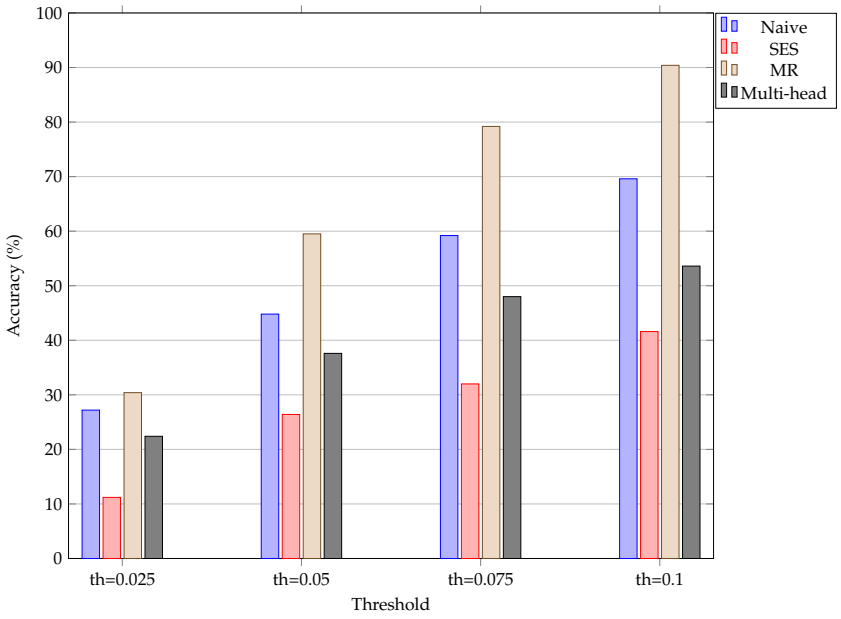


FIGURE A.129: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 2. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	22.4	37.6	48.0	53.6

TABLE A.51: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 2.*

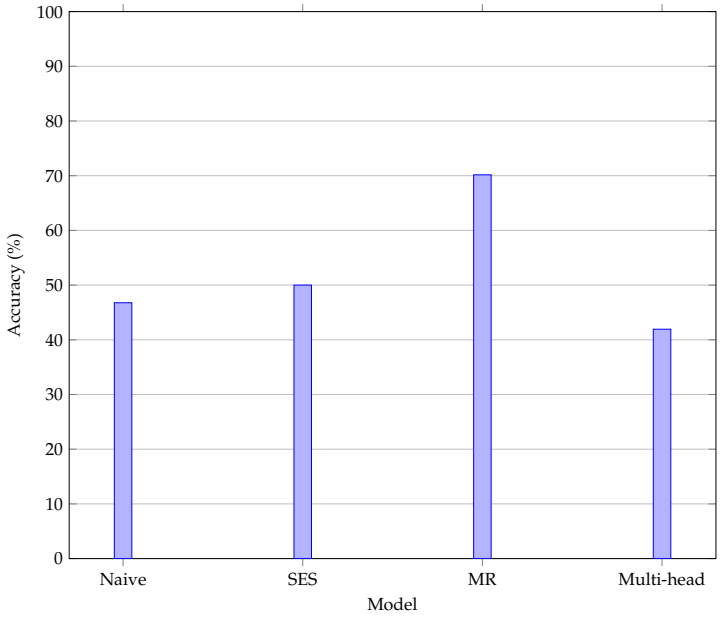


FIGURE A.130: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 2.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.52: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 2.*

A.3.3 Test 3 results

The results obtained for test 3 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.131 shows the predictions made by the Multi-head model,
- Figure A.132(a) and Figure A.132(b) show the change in accuracy and error as predictions are made,
- Figure A.133(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.133(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.134 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.135 shows the accuracy obtained when predicting the direction of price movement,
- Table A.53 represents the results obtained when the accuracy threshold is changed,
- Table A.54 illustrates the accuracy obtained when the direction of price movement is predicted.

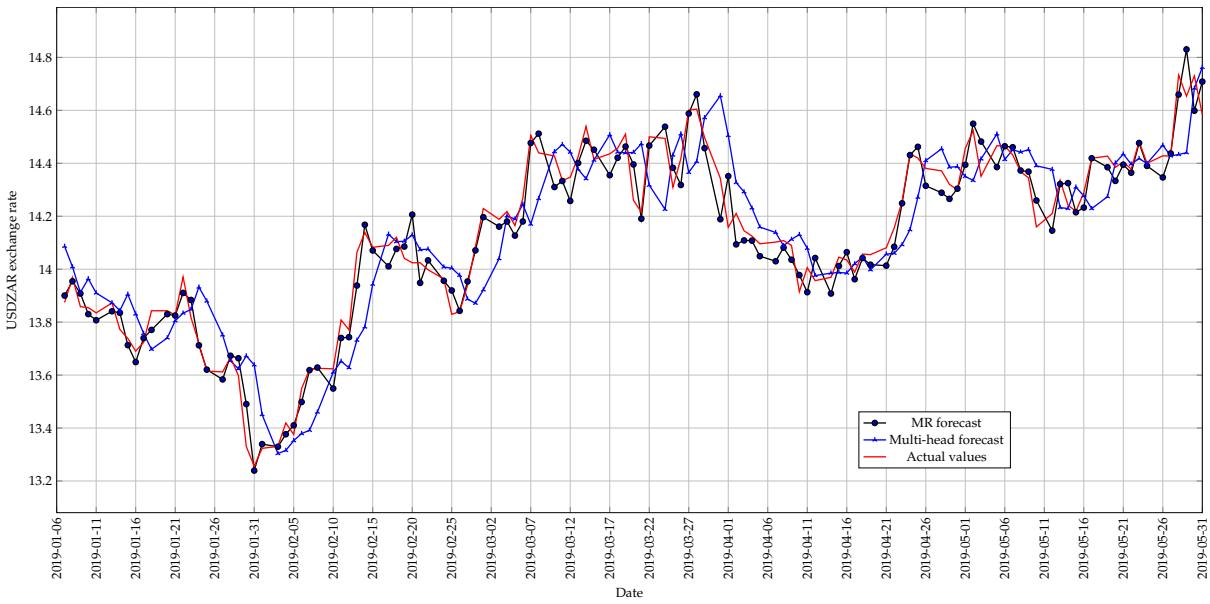
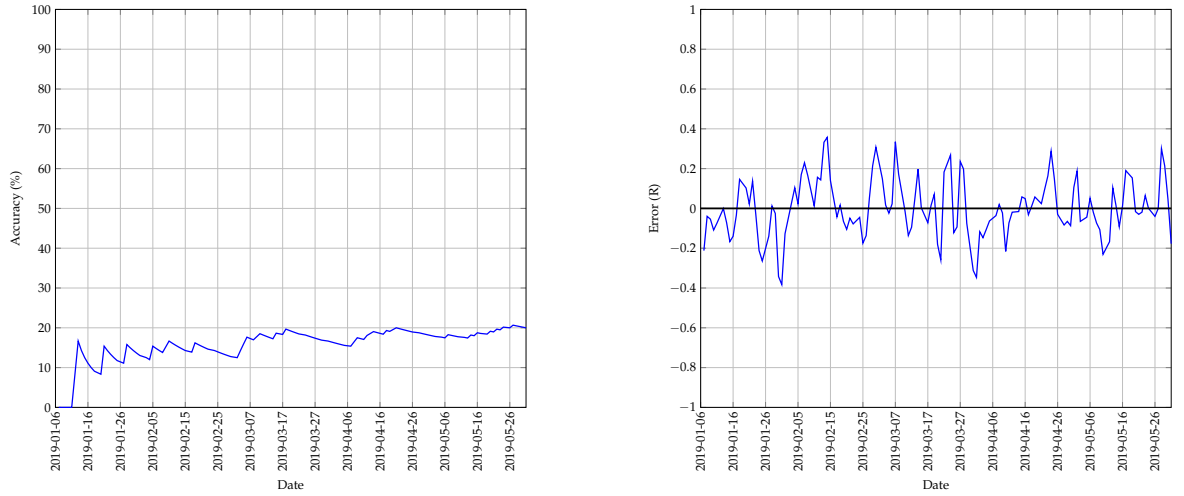


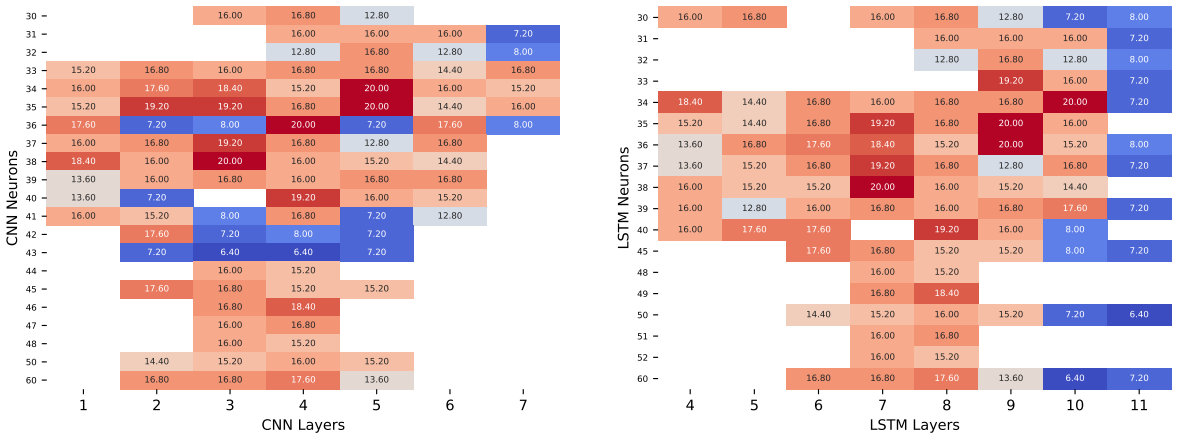
FIGURE A.131: *The best predictions made by the Multi-head model for test 3 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.132: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 3.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.133: *A graphical representation of the influence that different architectures have on prediction accuracy for test 3 of the Multi-head model.*

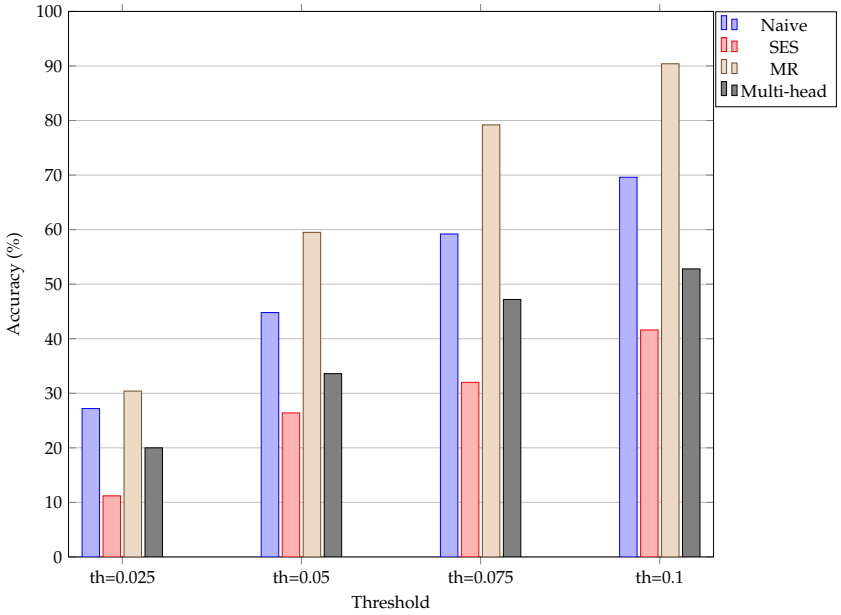


FIGURE A.134: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 3. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	20.0	33.6	47.2	52.8

TABLE A.53: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 3.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.54: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 3.*

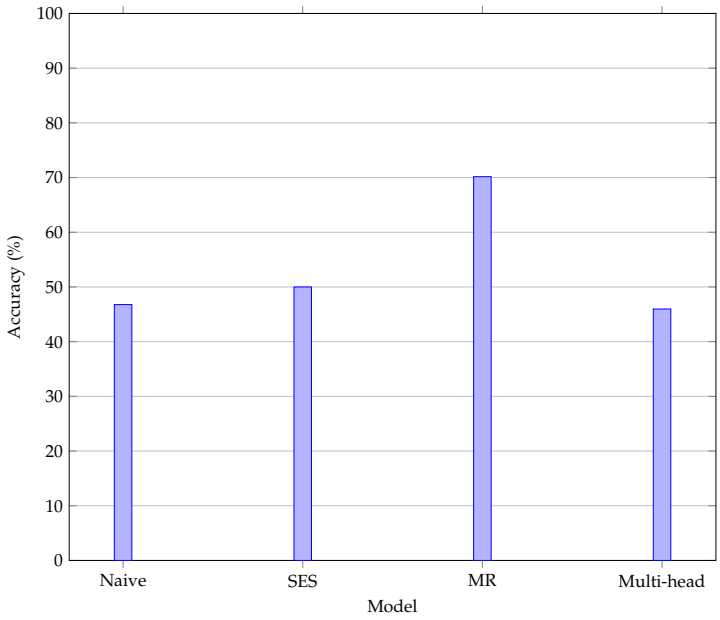


FIGURE A.135: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 3.*

A.3.4 Test 4 results

The results obtained for test 4 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.136 shows the predictions made by the Multi-head model,
- Figure A.137(a) and Figure A.137(b) show the change in accuracy and error as predictions are made,
- Figure A.138(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.138(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.139 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.140 shows the accuracy obtained when predicting the direction of price movement,
- Table A.55 represents the results obtained when the accuracy threshold is changed,
- Table A.56 illustrates the accuracy obtained when the direction of price movement is predicted.

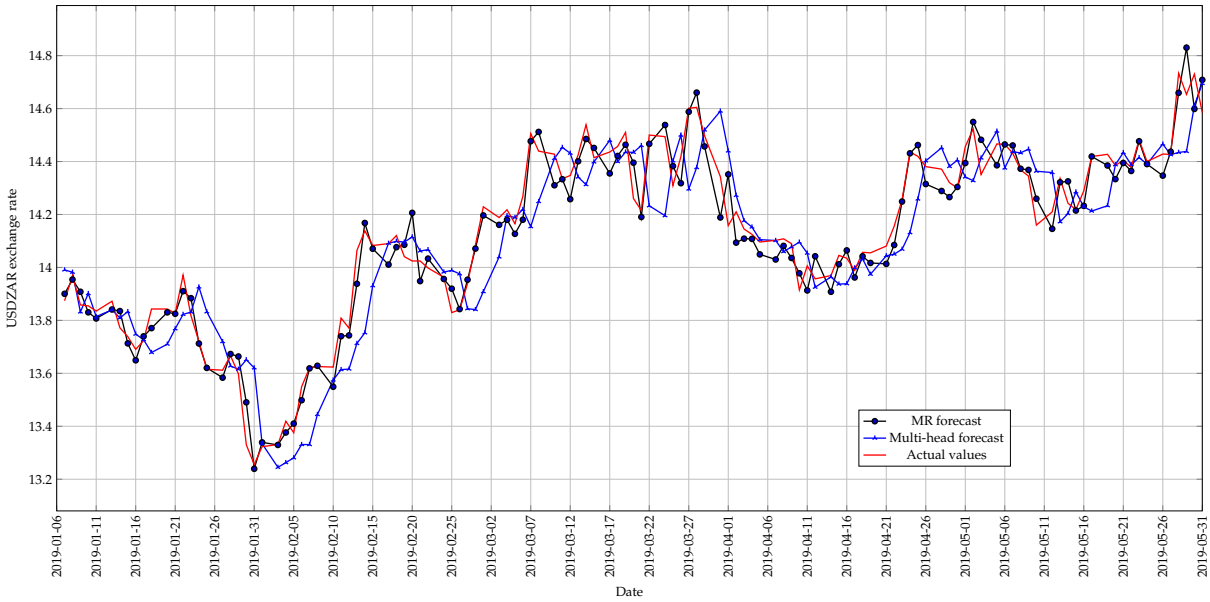
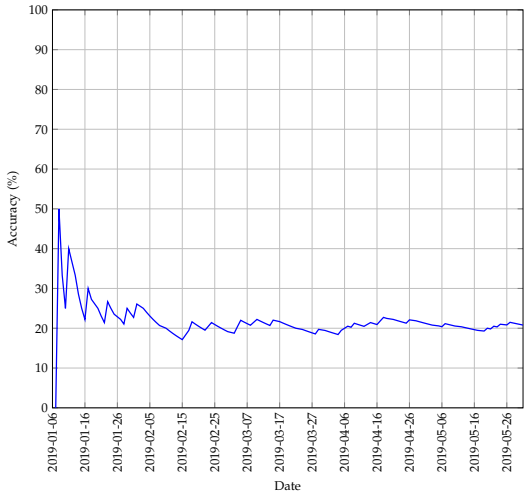
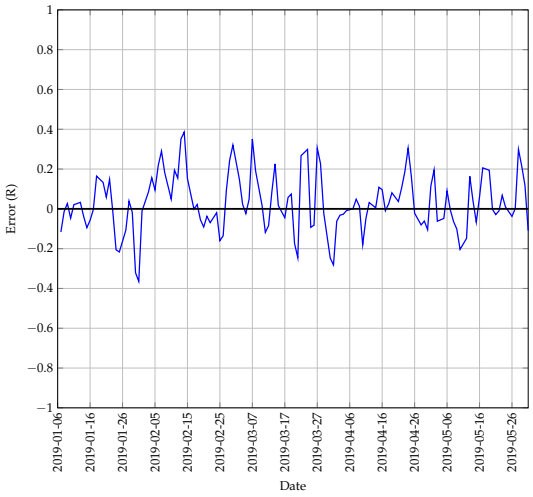


FIGURE A.136: *The best predictions made by the Multi-head model for test 4 with 1 previous day as input using a walk-forward validation approach on the test set.*

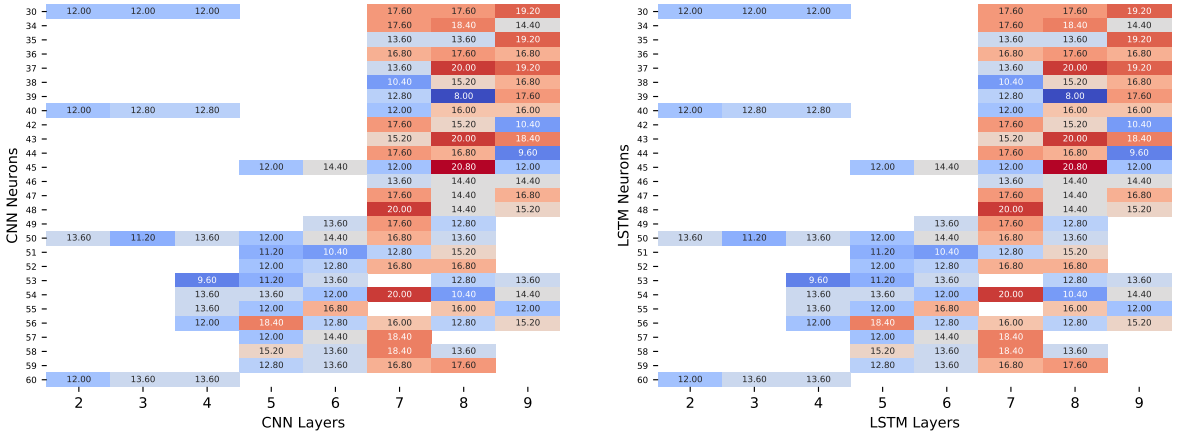


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.137: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 4.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.138: *A graphical representation of the influence that different architectures have on prediction accuracy for test 4 of the Multi-head model.*

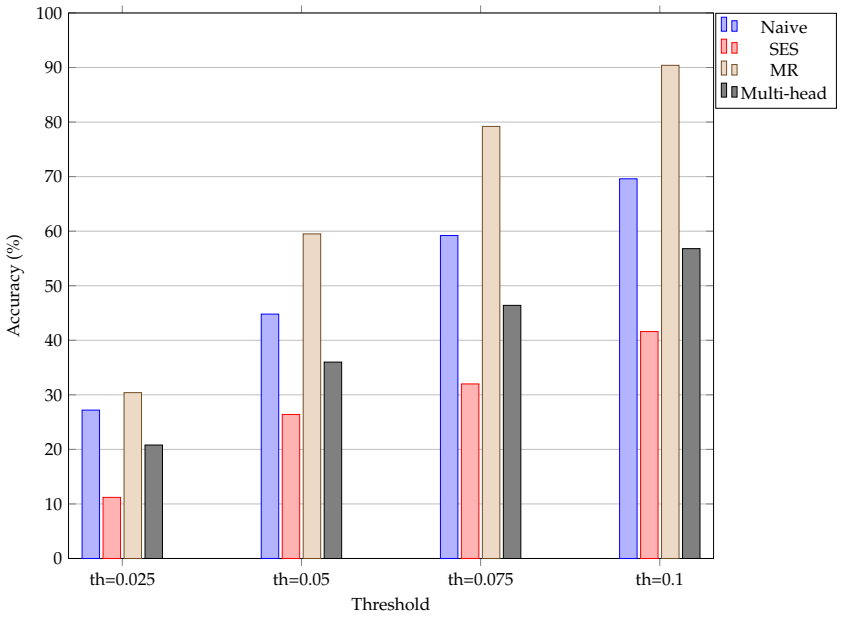


FIGURE A.139: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 4. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	20.8	36.0	46.4	56.8

TABLE A.55: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 4.*

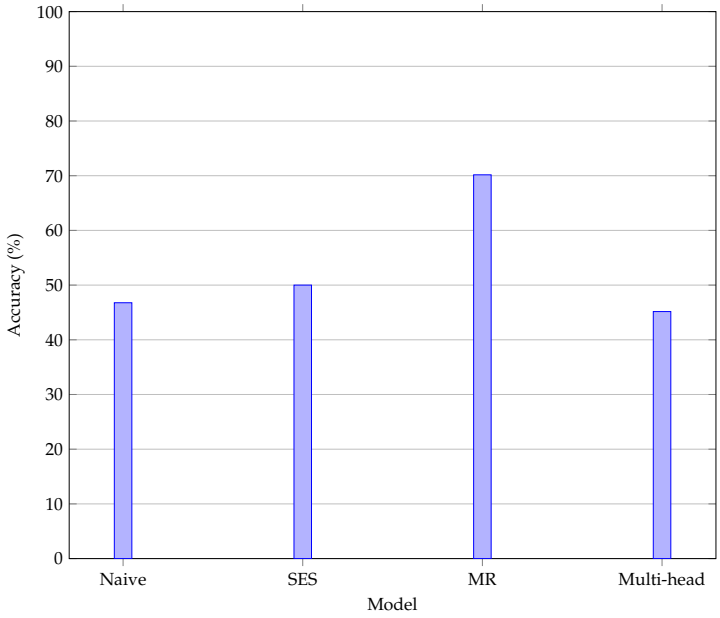


FIGURE A.140: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 4.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.56: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 4.*

A.3.5 Test 5 results

The results obtained for test 5 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.141 shows the predictions made by the Multi-head model,
- Figure A.142(a) and Figure A.142(b) show the change in accuracy and error as predictions are made,
- Figure A.143(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.143(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.144 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.145 shows the accuracy obtained when predicting the direction of price movement,
- Table A.57 represents the results obtained when the accuracy threshold is changed,
- Table A.58 illustrates the accuracy obtained when the direction of price movement is predicted.

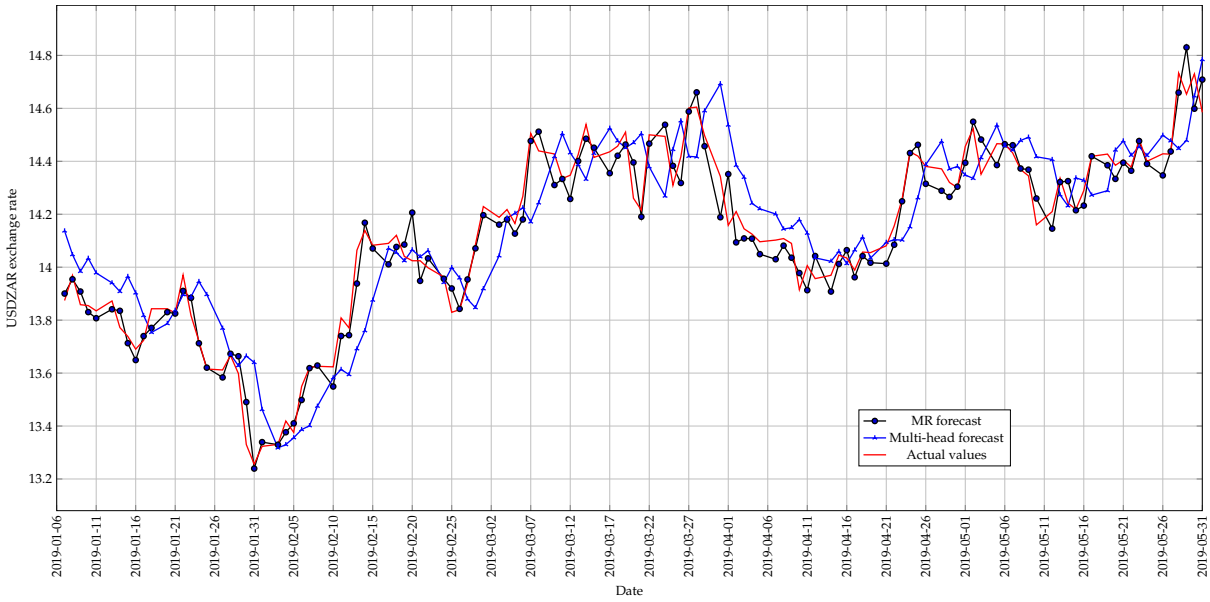
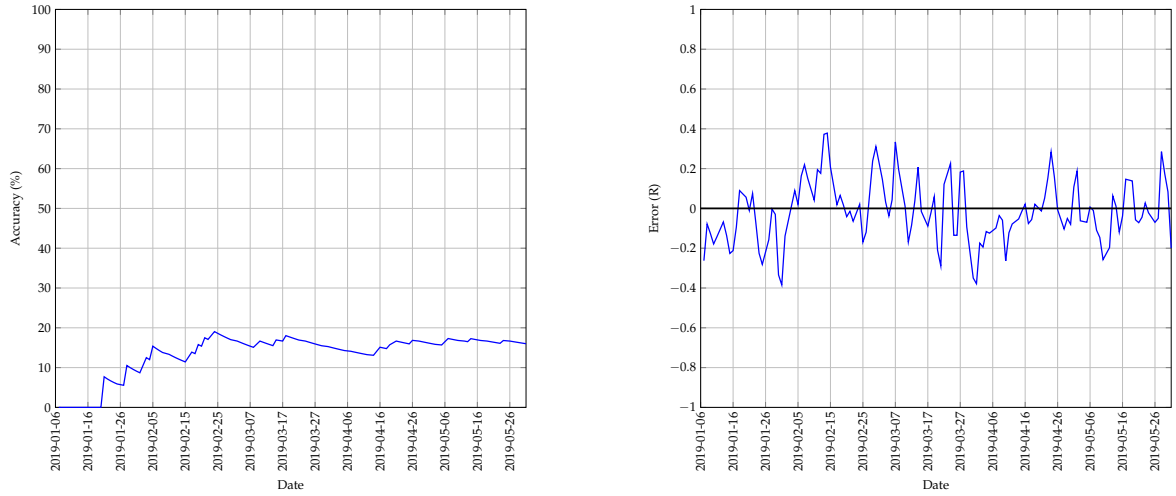


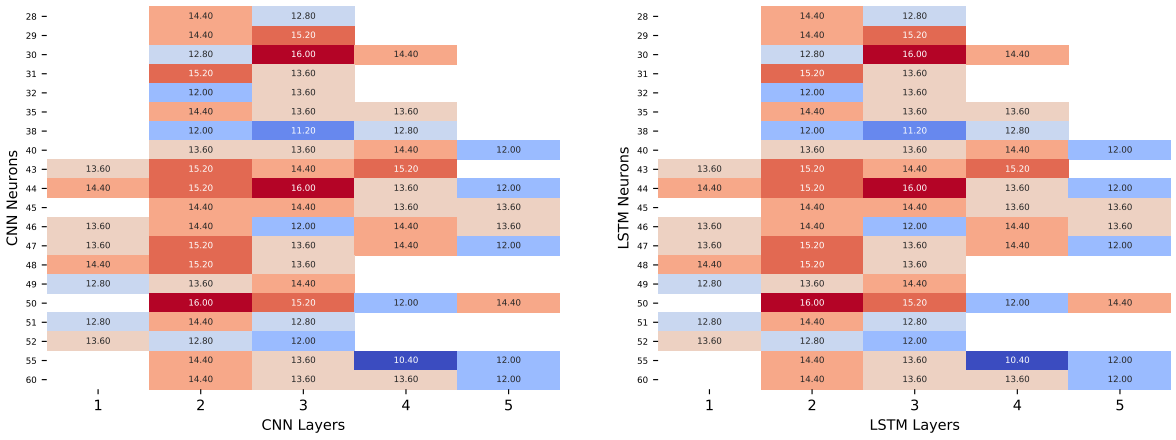
FIGURE A.141: *The best predictions made by the Multi-head model for test 5 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.142: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 5.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.143: *A graphical representation of the influence that different architectures have on prediction accuracy for test 5 of the Multi-head model.*

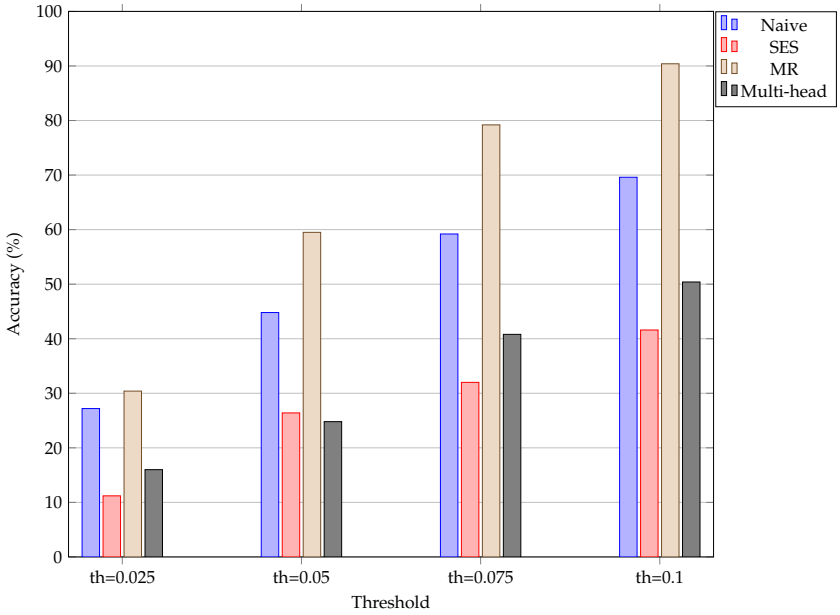


FIGURE A.144: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 5. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	16.0	24.8	40.8	50.4

TABLE A.57: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 5.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.58: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 5.*

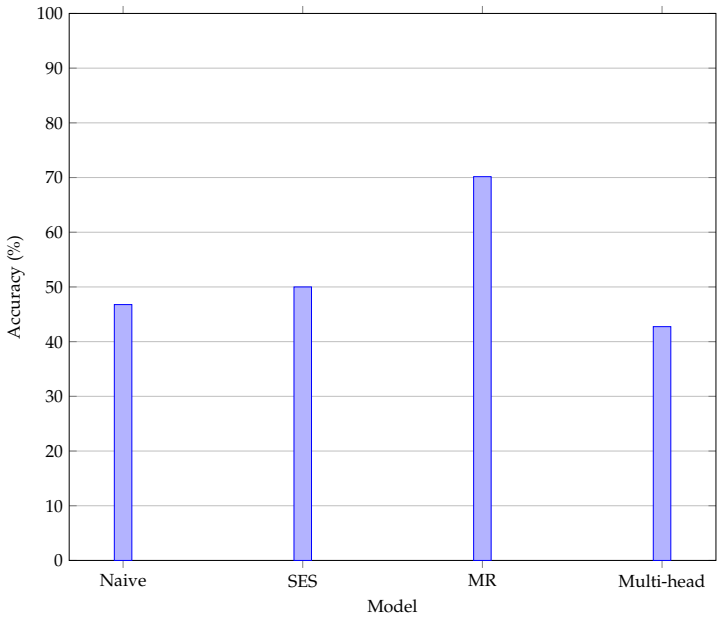


FIGURE A.145: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 5.*

A.3.6 Test 6 results

The results obtained for test 6 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.146 shows the predictions made by the Multi-head model,
- Figure A.147(a) and Figure A.147(b) show the change in accuracy and error as predictions are made,
- Figure A.148(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.148(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.149 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.150 shows the accuracy obtained when predicting the direction of price movement,
- Table A.59 represents the results obtained when the accuracy threshold is changed,
- Table A.60 illustrates the accuracy obtained when the direction of price movement is predicted.

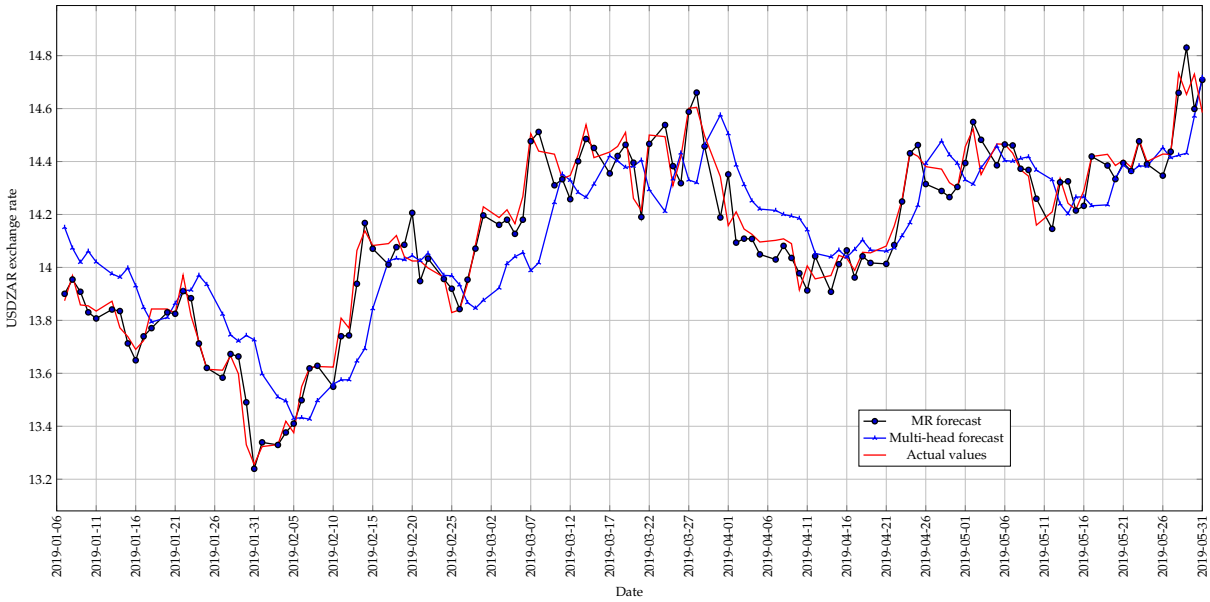
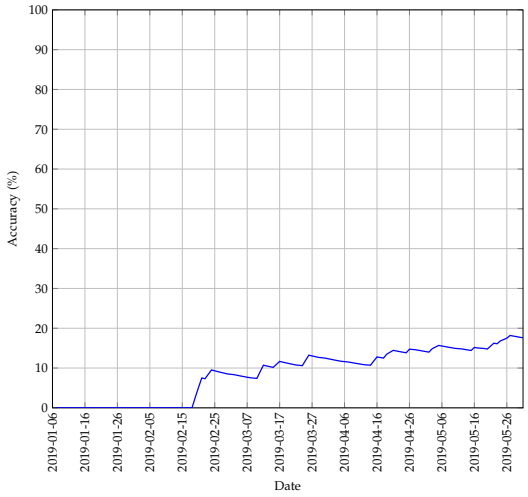
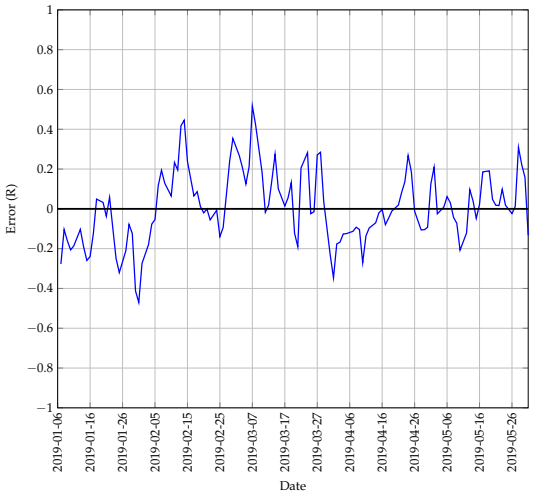


FIGURE A.146: *The best predictions made by the Multi-head model for test 6 with 1 previous day as input using a walk-forward validation approach on the test set.*

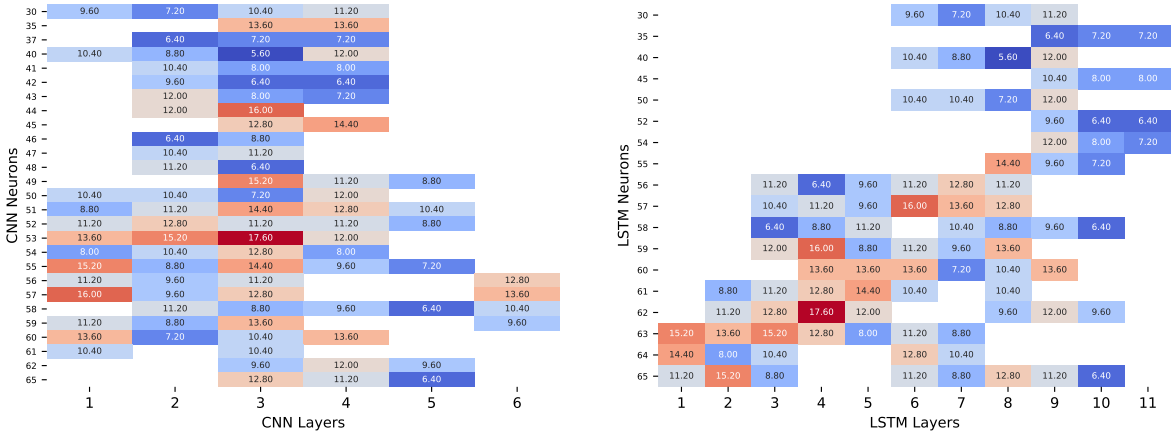


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.147: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 6.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.148: *A graphical representation of the influence that different architectures have on prediction accuracy for test 6 of the Multi-head model.*

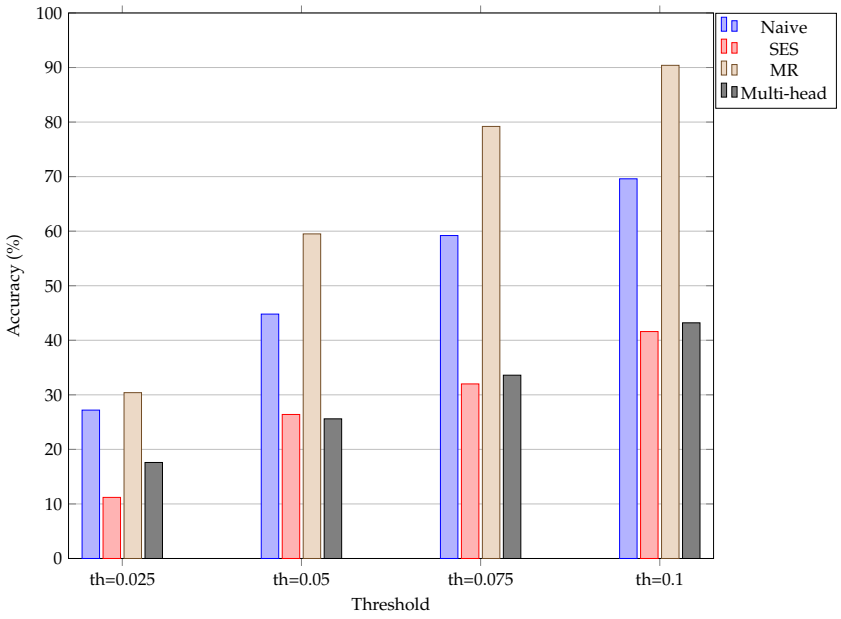


FIGURE A.149: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 6. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	17.6	25.6	33.6	43.2

TABLE A.59: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 6.*

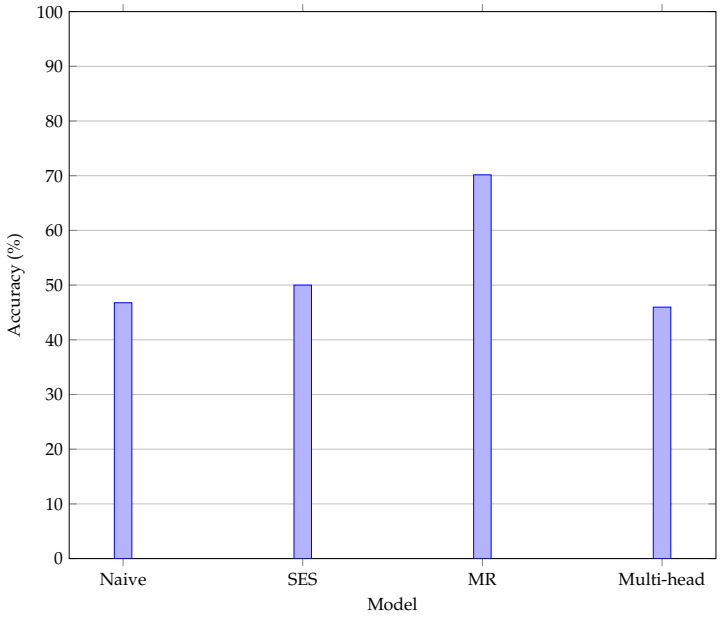


FIGURE A.150: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 6.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.60: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 6.*

A.3.7 Test 7 results

The results obtained for test 7 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.151 shows the predictions made by the Multi-head model,
- Figure A.152(a) and Figure A.152(b) show the change in accuracy and error as predictions are made,
- Figure A.153(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.153(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.154 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.155 shows the accuracy obtained when predicting the direction of price movement,
- Table A.61 represents the results obtained when the accuracy threshold is changed,
- Table A.62 illustrates the accuracy obtained when the direction of price movement is predicted.

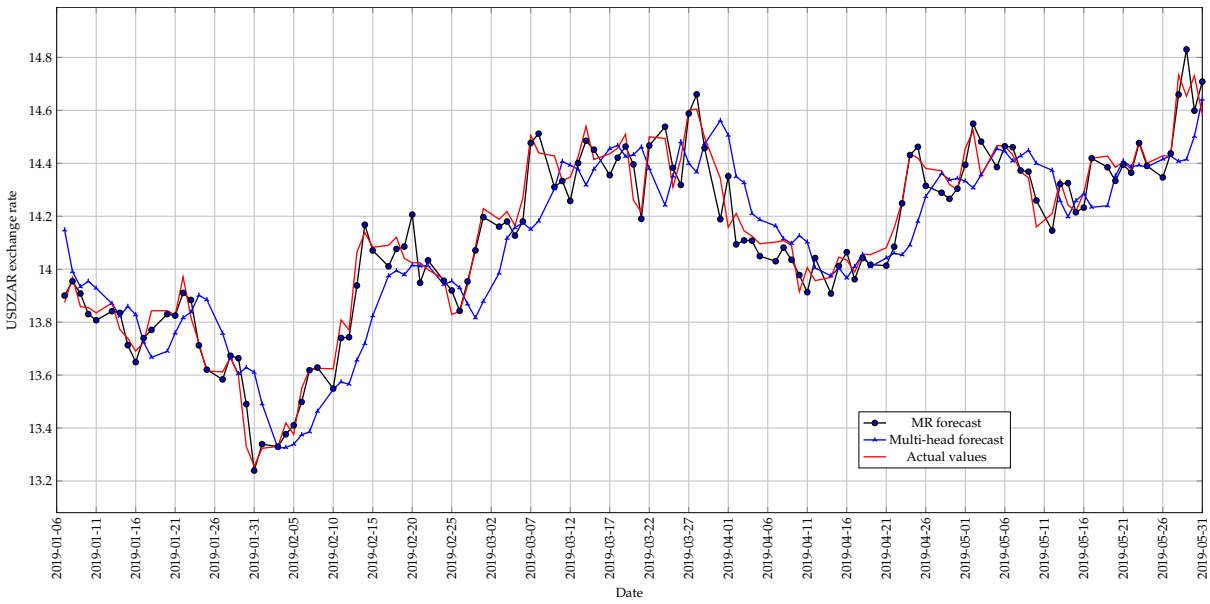
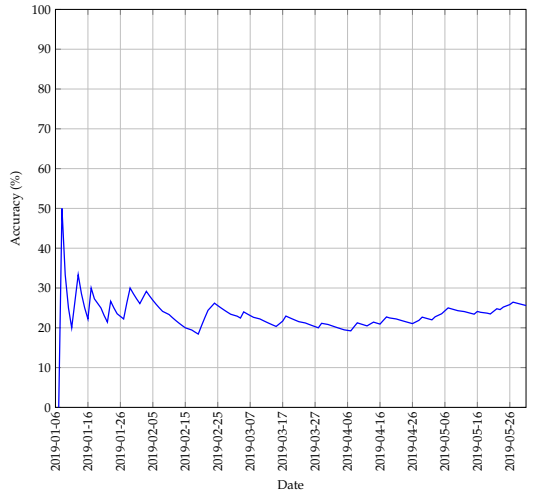
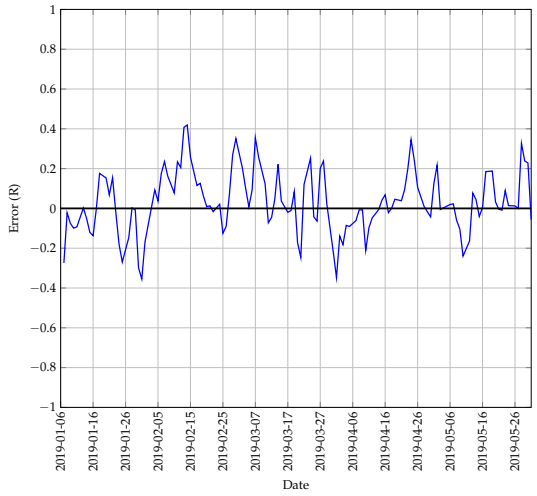


FIGURE A.151: *The best predictions made by the Multi-head model for test 7 with 1 previous day as input using a walk-forward validation approach on the test set.*

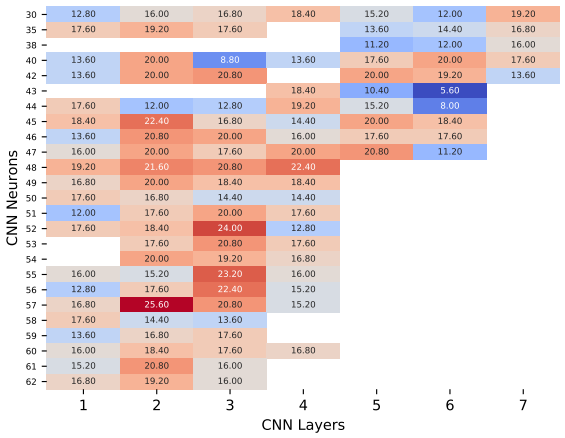


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

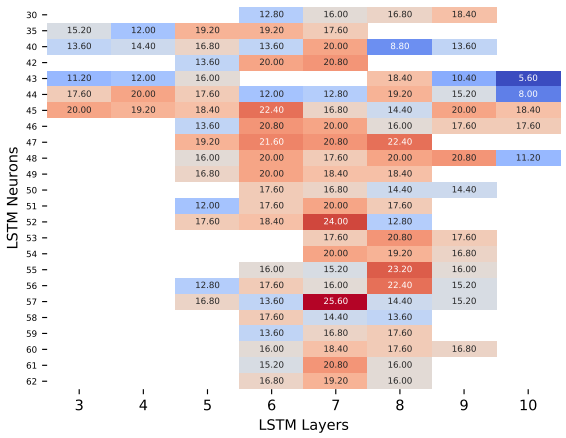


(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.152: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 7.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*



(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.153: *A graphical representation of the influence that different architectures have on prediction accuracy for test 7 of the Multi-head model.*

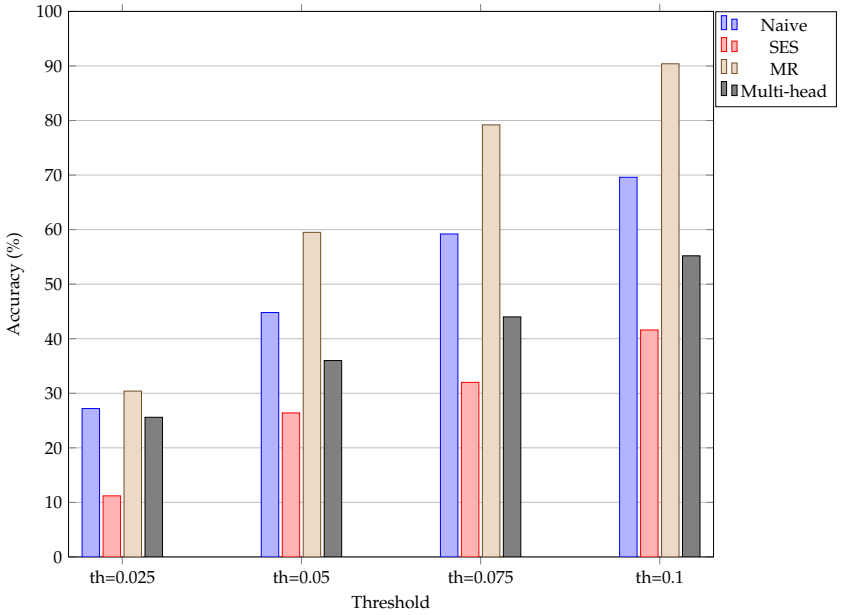


FIGURE A.154: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 7. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	25.6	36.0	44.0	55.2

TABLE A.61: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 7.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.62: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 7.*

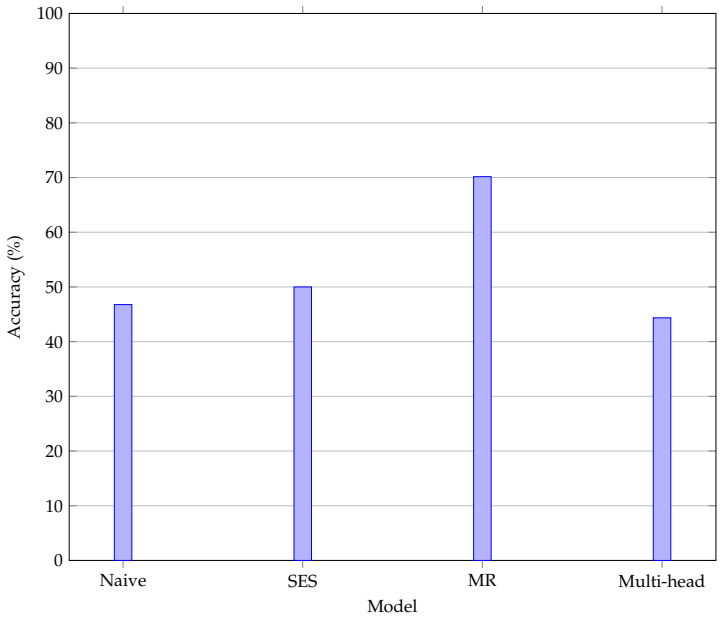


FIGURE A.155: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 7.*

A.3.8 Test 8 results

The results obtained for test 8 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.156 shows the predictions made by the Multi-head model,
- Figure A.157(a) and Figure A.157(b) show the change in accuracy and error as predictions are made,
- Figure A.158(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.158(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.159 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.160 shows the accuracy obtained when predicting the direction of price movement,
- Table A.63 represents the results obtained when the accuracy threshold is changed,
- Table A.64 illustrates the accuracy obtained when the direction of price movement is predicted.

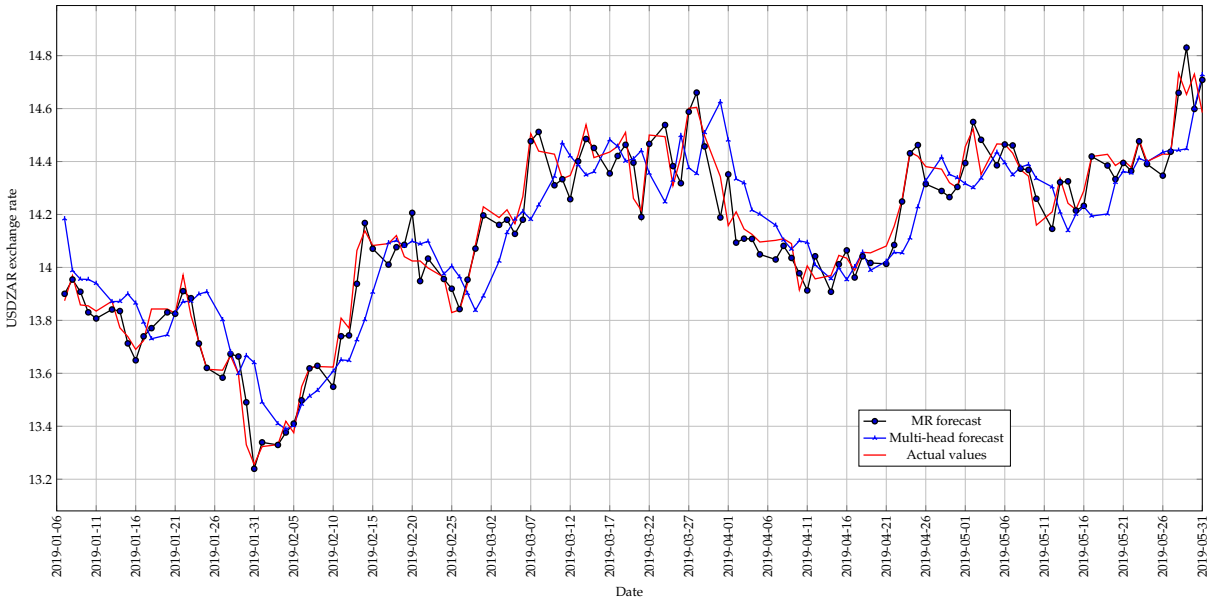
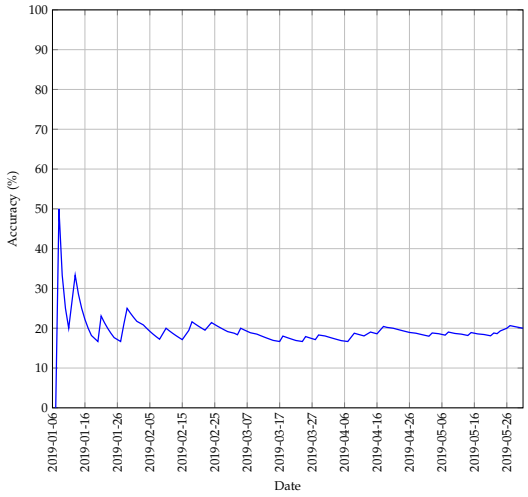
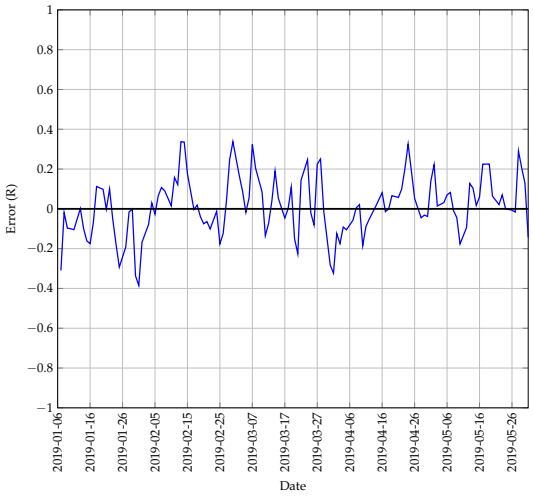


FIGURE A.156: *The best predictions made by the Multi-head model for test 8 with 1 previous day as input using a walk-forward validation approach on the test set.*

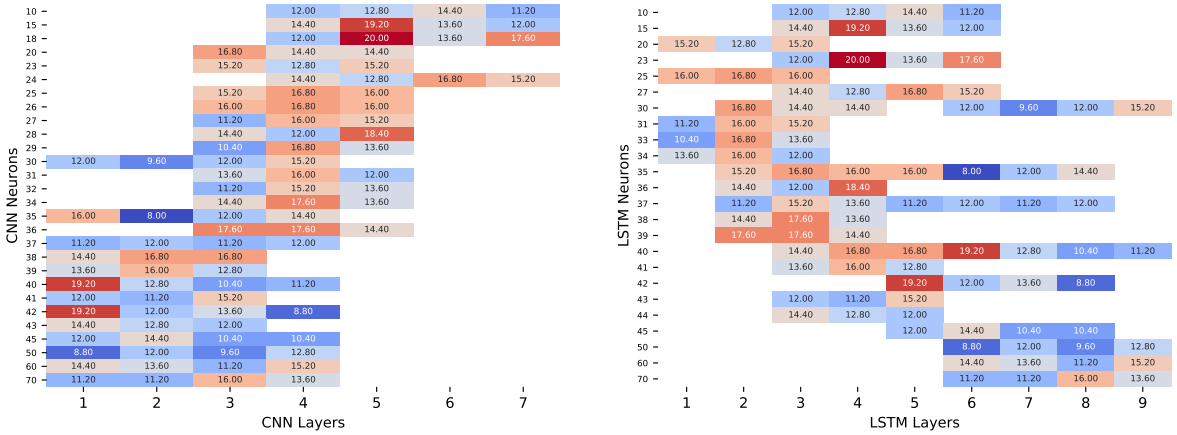


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.157: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 8.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.158: *A graphical representation of the influence that different architectures have on prediction accuracy for test 8 of the Multi-head model.*

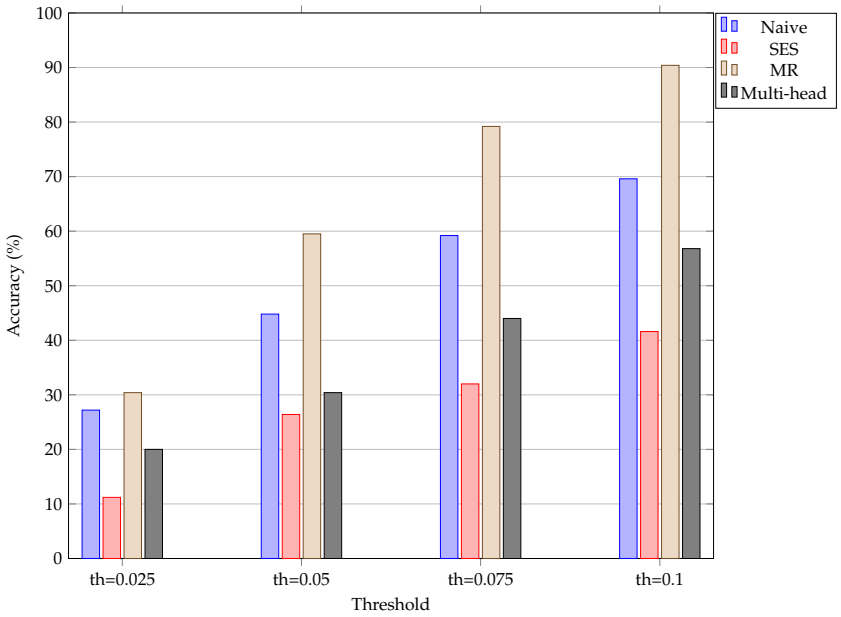


FIGURE A.159: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 8. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	20.0	30.4	44.0	56.8

TABLE A.63: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 8.*

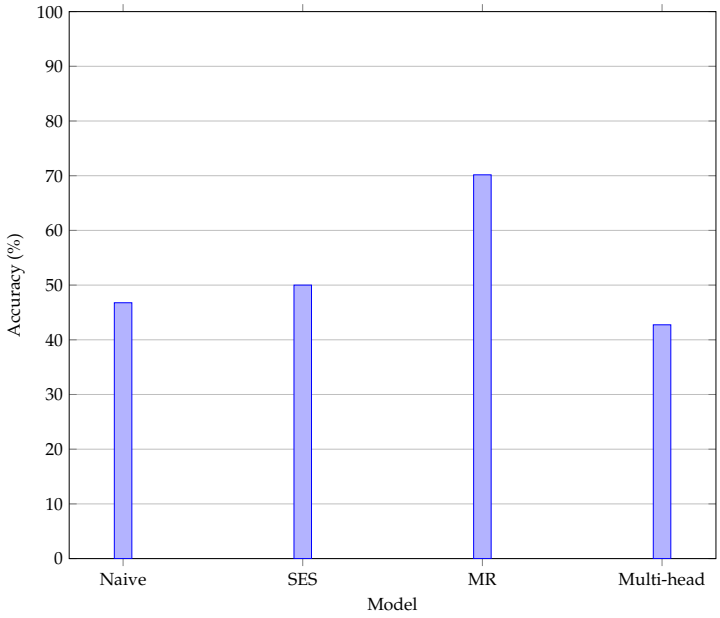


FIGURE A.160: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 8.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.64: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 8.*

A.3.9 Test 9 results

The results obtained for test 9 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.161 shows the predictions made by the Multi-head model,
- Figure A.162(a) and Figure A.162(b) show the change in accuracy and error as predictions are made,
- Figure A.163(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.163(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.164 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.165 shows the accuracy obtained when predicting the direction of price movement,
- Table A.65 represents the results obtained when the accuracy threshold is changed,
- Table A.66 illustrates the accuracy obtained when the direction of price movement is predicted.

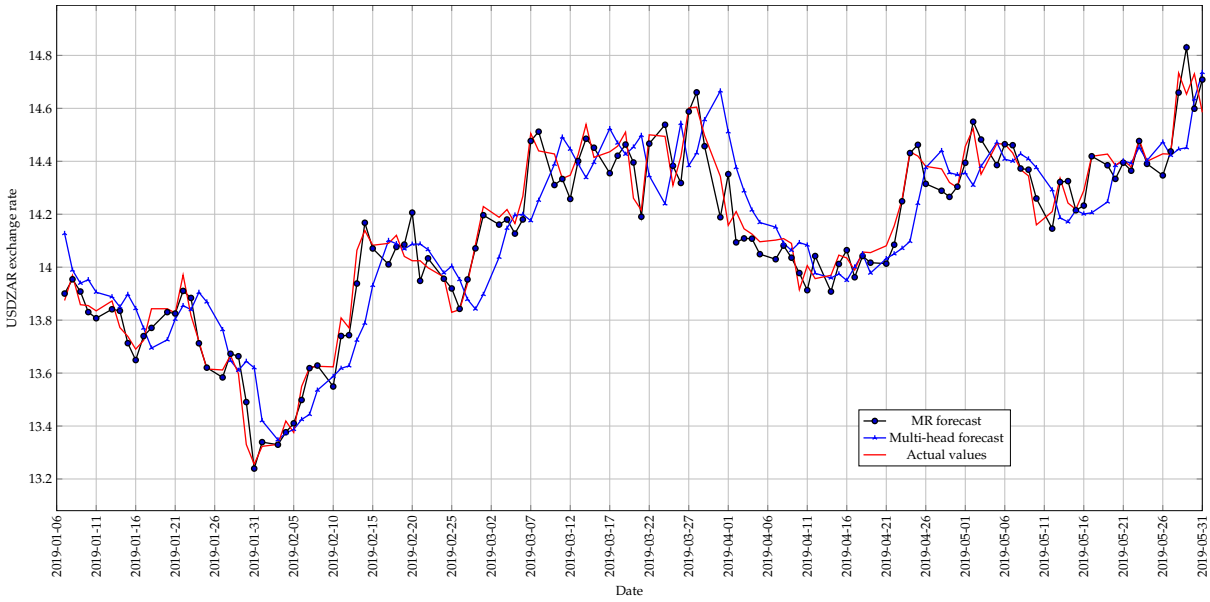
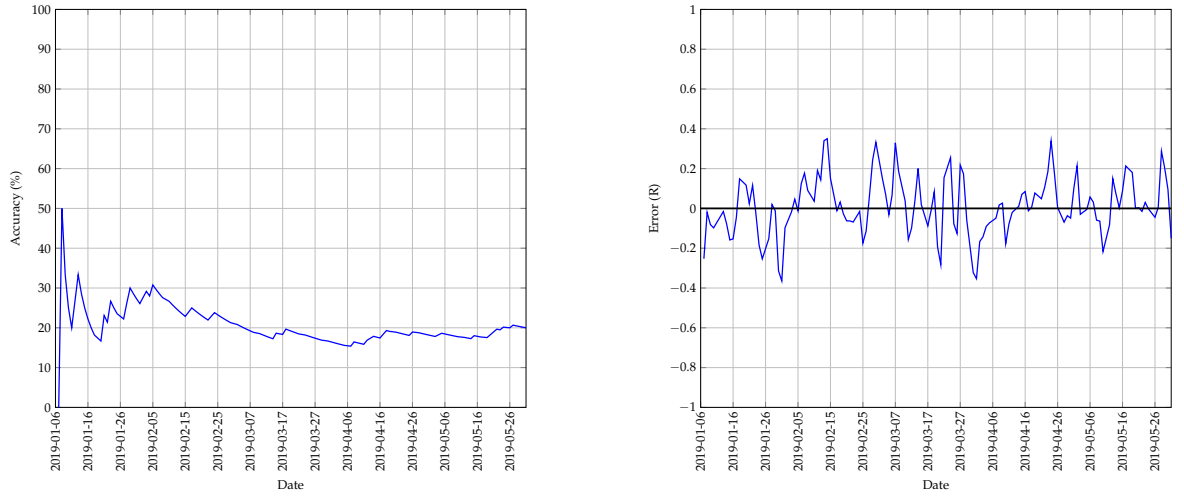


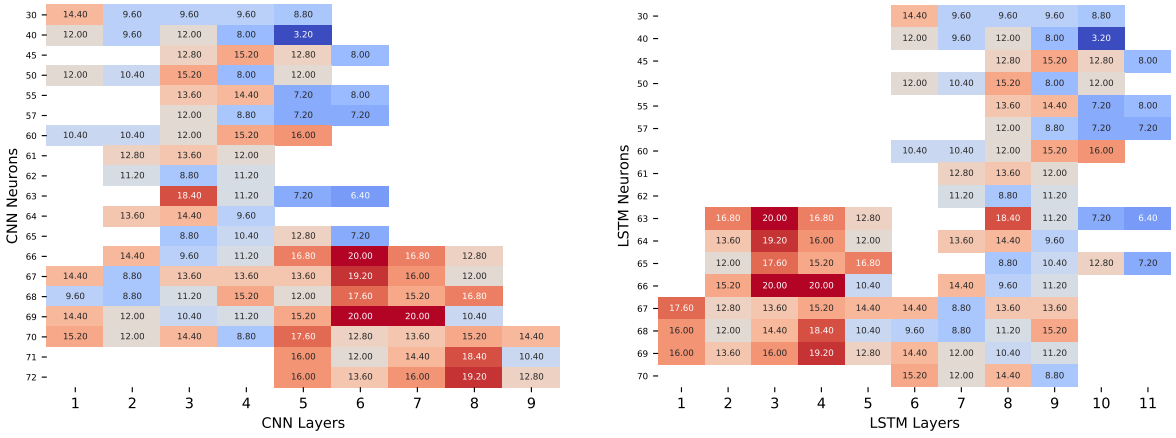
FIGURE A.161: *The best predictions made by the Multi-head model for test 9 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.162: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 9.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.163: *A graphical representation of the influence that different architectures have on prediction accuracy for test 9 of the Multi-head model.*

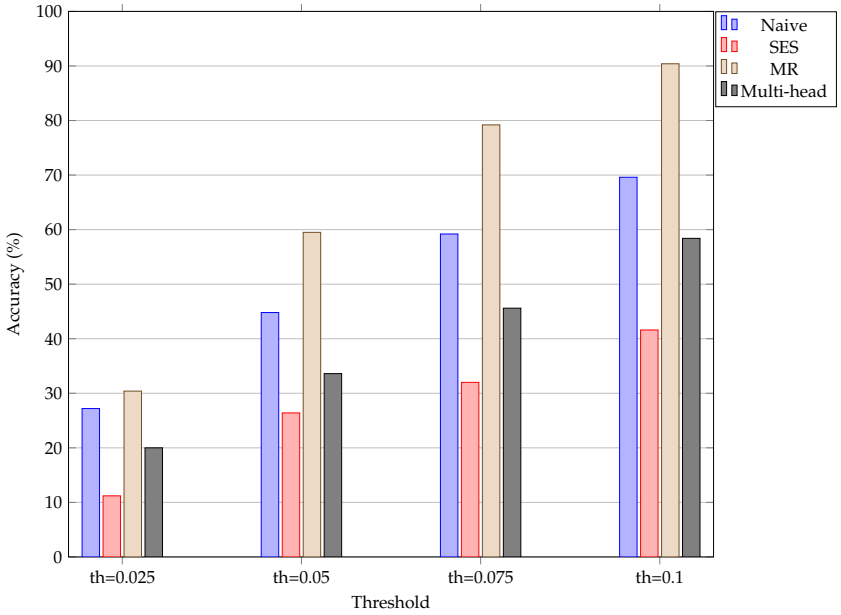


FIGURE A.164: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 9. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	20.0	33.6	45.6	58.4

TABLE A.65: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 9.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.66: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 9.*

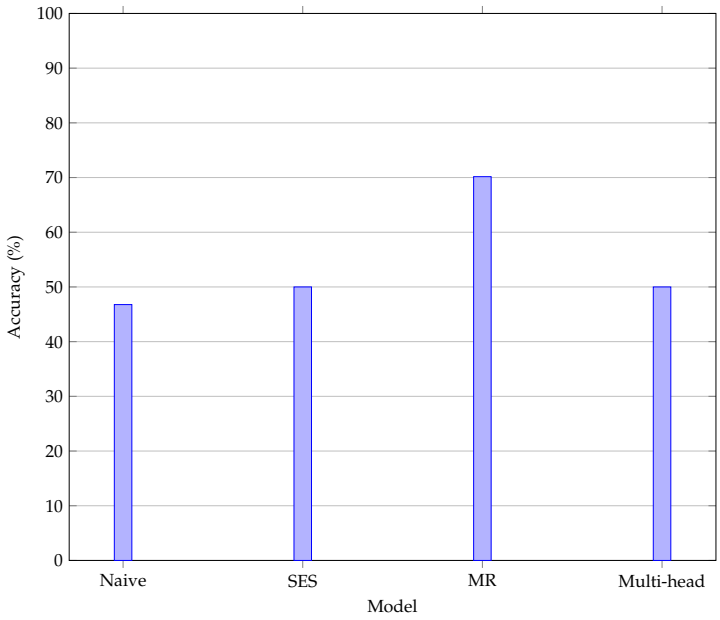


FIGURE A.165: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 9.*

A.3.10 Test 10 results

The results obtained for test 10 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.166 shows the predictions made by the Multi-head model,
- Figure A.167(a) and Figure A.167(b) show the change in accuracy and error as predictions are made,
- Figure A.168(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.168(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.169 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.170 shows the accuracy obtained when predicting the direction of price movement,
- Table A.67 represents the results obtained when the accuracy threshold is changed,
- Table A.68 illustrates the accuracy obtained when the direction of price movement is predicted.

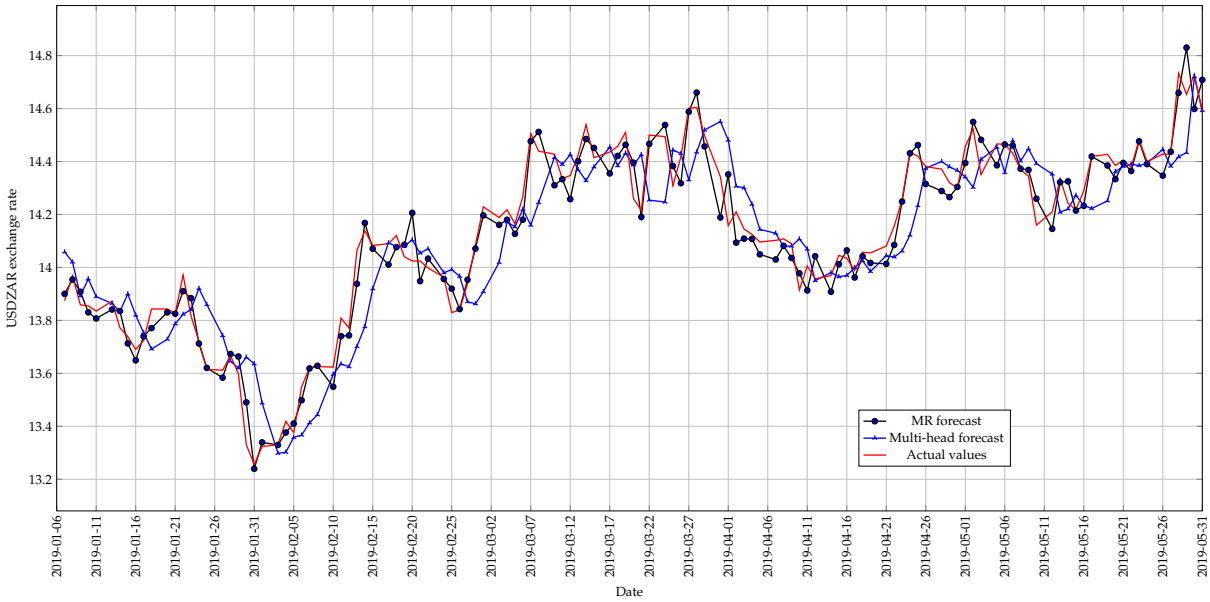
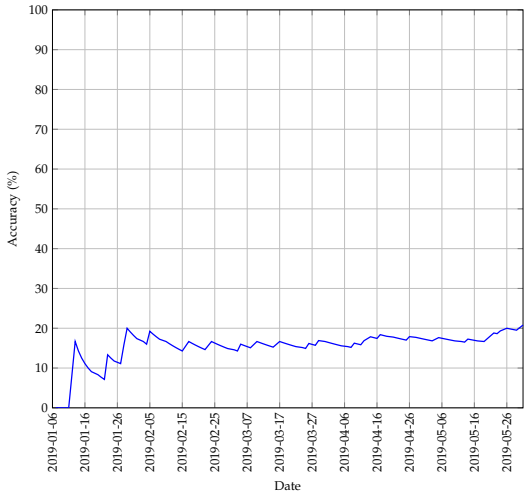
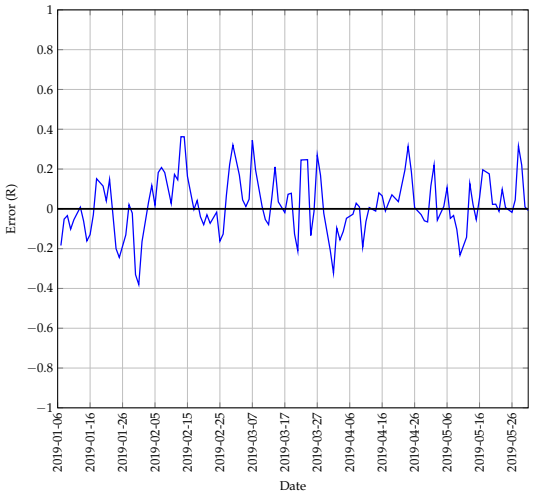


FIGURE A.166: *The best predictions made by the Multi-head model for test 10 with 1 previous day as input using a walk-forward validation approach on the test set.*

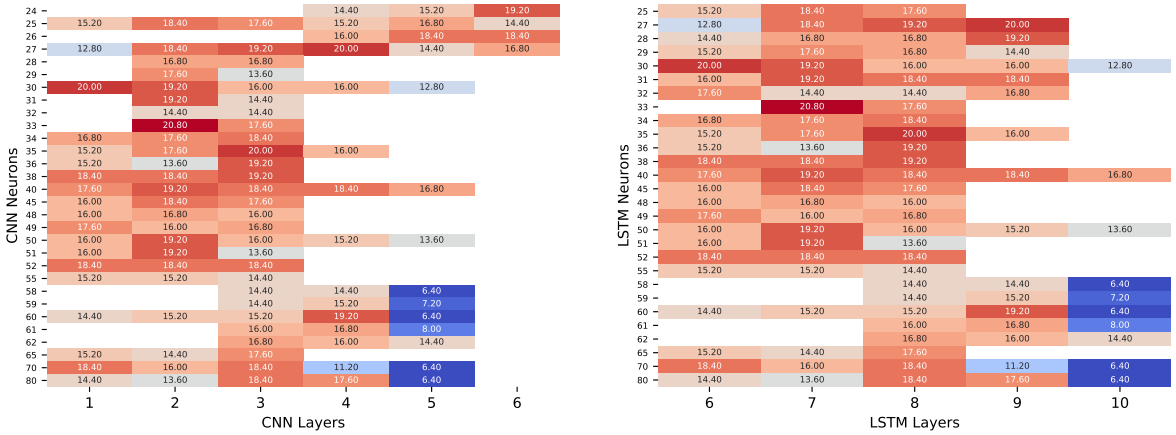


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.167: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 10.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.168: *A graphical representation of the influence that different architectures have on prediction accuracy for test 10 of the Multi-head model.*

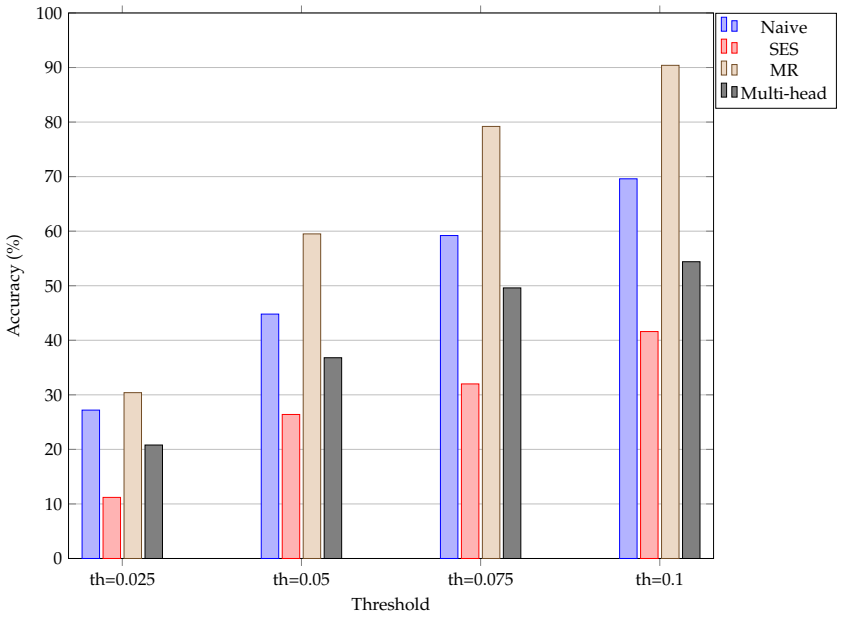


FIGURE A.169: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 10. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	20.8	36.8	49.6	54.4

TABLE A.67: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 10.*

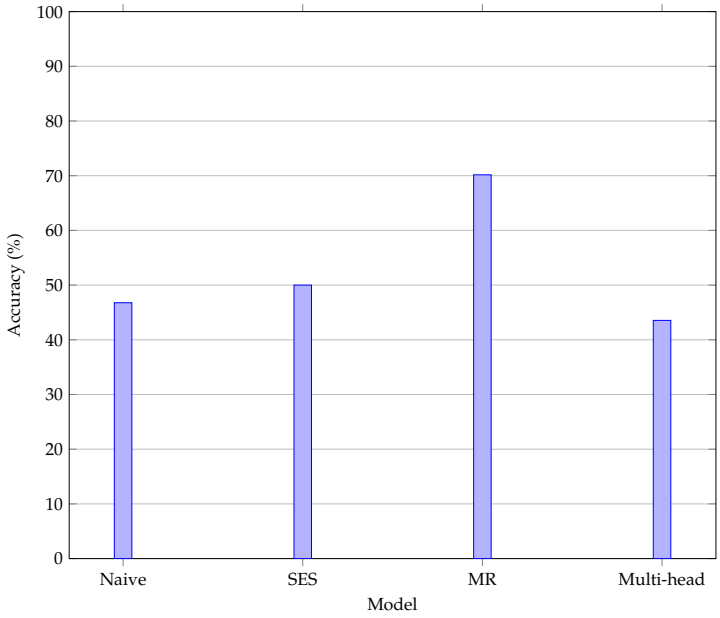


FIGURE A.170: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 10.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.68: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 10.*

A.3.11 Test 11 results

The results obtained for test 11 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.171 shows the predictions made by the Multi-head model,
- Figure A.172(a) and Figure A.172(b) show the change in accuracy and error as predictions are made,
- Figure A.173(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.173(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.174 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.175 shows the accuracy obtained when predicting the direction of price movement,
- Table A.69 represents the results obtained when the accuracy threshold is changed,
- Table A.70 illustrates the accuracy obtained when the direction of price movement is predicted.

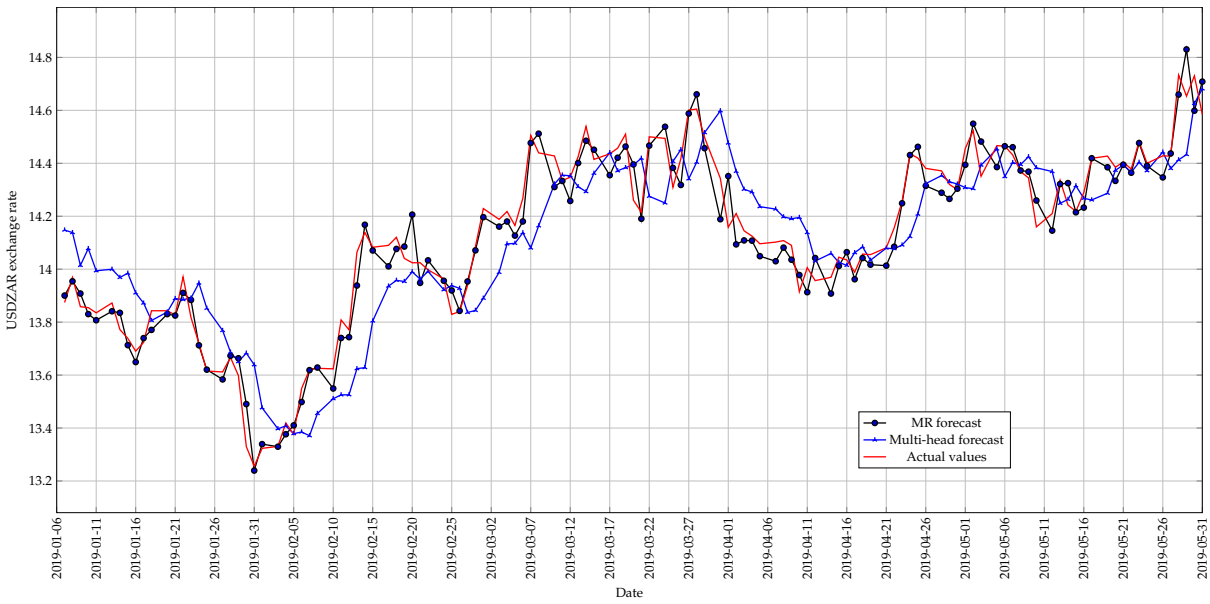
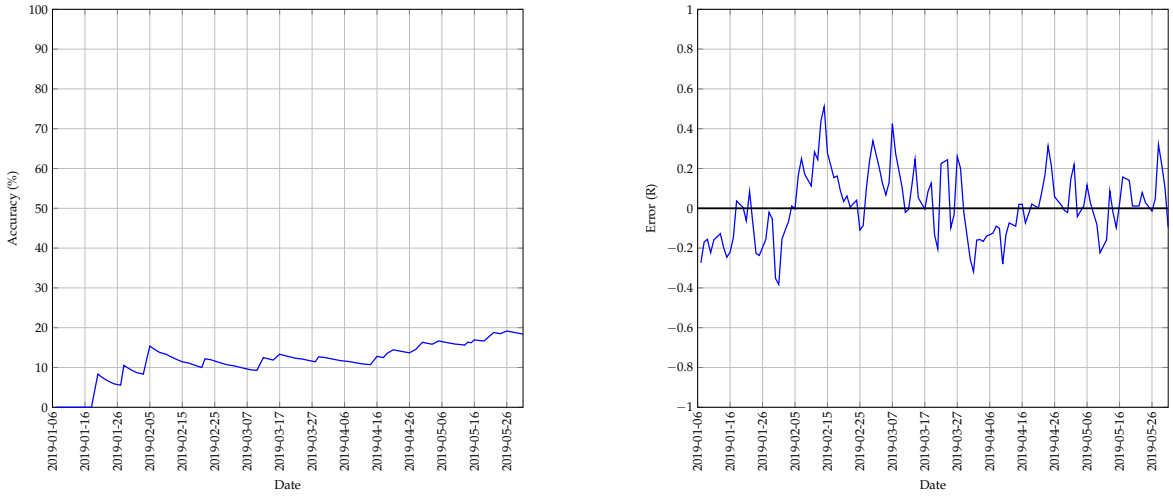


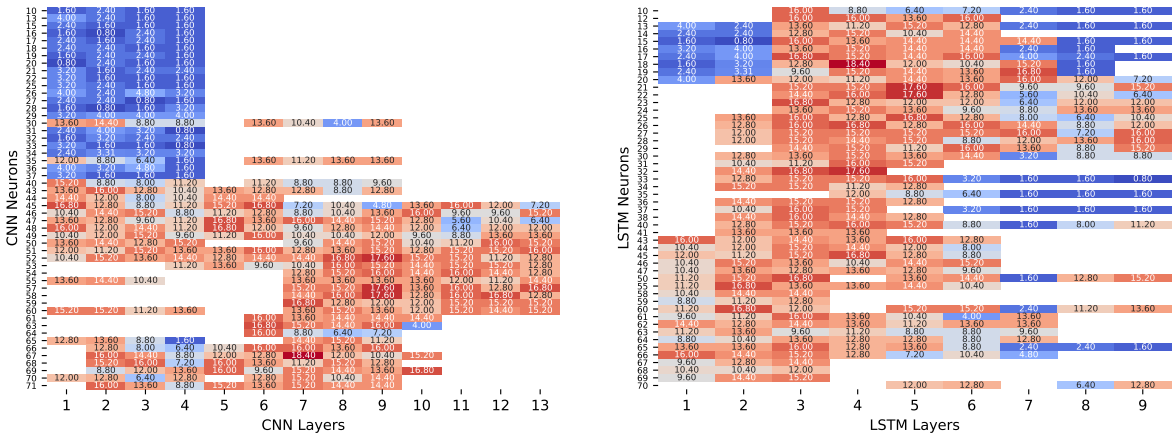
FIGURE A.171: *The best predictions made by the Multi-head model for test 11 with 1 previous day as input using a walk-forward validation approach on the test set.*



(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*

(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.172: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 11.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 1 previous day as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 1 previous day as input.*

FIGURE A.173: *A graphical representation of the influence that different architectures have on prediction accuracy for test 11 of the Multi-head model.*

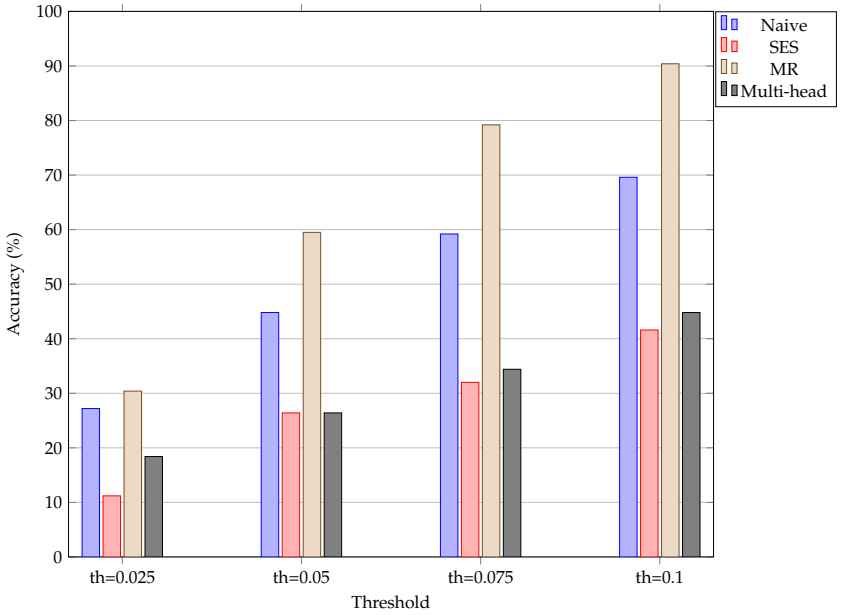


FIGURE A.174: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 11. The "th=0.025" represents a threshold set at R0.025.*

Model	**Threshold of R0.025 (%)**	**Threshold of R0.05 (%)**	**Threshold of R0.075 (%)**	**Threshold of R0.1 (%)**
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	18.4	26.4	34.4	44.8

TABLE A.69: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 11.*

Model	**Accuracy (%)**
Naive	46.774
SES	50.000
MR	70.161
Multi-head	46.774

TABLE A.70: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 11.*

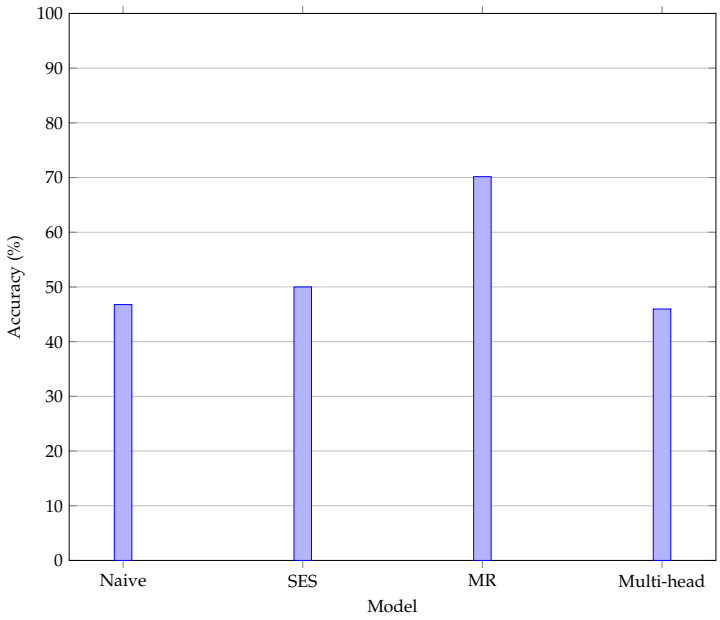


FIGURE A.175: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 11.*

A.3.12 Test 12 results

The results obtained for test 12 of the Multi-head model are shown in this section. The figures and tables shown are summarised as follows:

- Figure A.176 shows the predictions made by the Multi-head model,
- Figure A.177(a) and Figure A.177(b) show the change in accuracy and error as predictions are made,
- Figure A.178(a) shows the accuracy obtained for different neural network architectures for the CNN,
- Figure A.178(b) shows the accuracy obtained for different neural network architectures for the LSTM,
- Figure A.179 illustrates the change in accuracy when the accuracy threshold is changed incrementally,
- Figure A.180 shows the accuracy obtained when predicting the direction of price movement,
- Table A.71 represents the results obtained when the accuracy threshold is changed,
- Table A.72 illustrates the accuracy obtained when the direction of price movement is predicted.

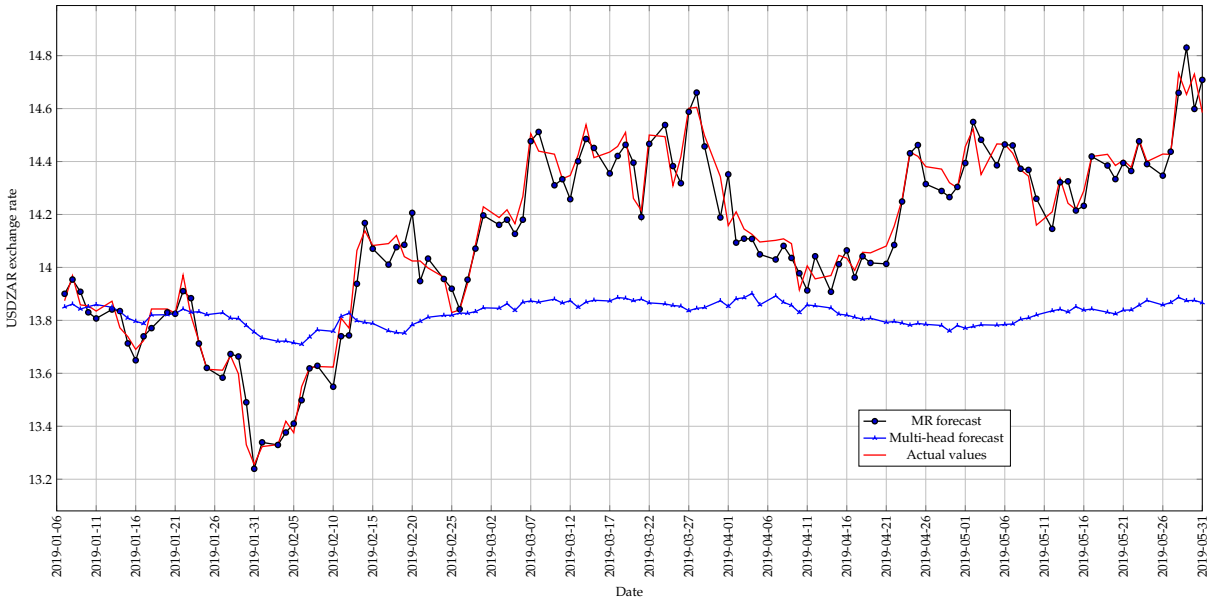
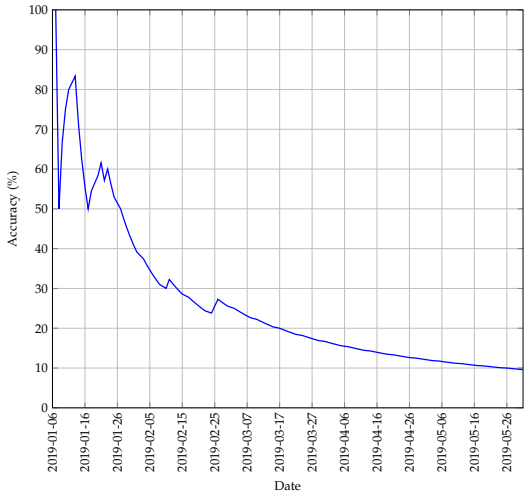
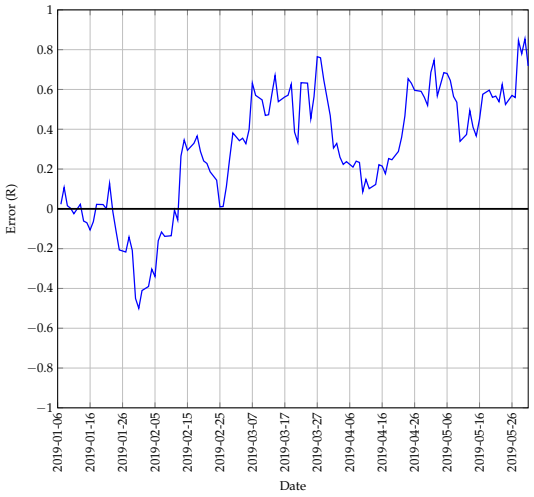


FIGURE A.176: *The best predictions made by the Multi-head model for test 12 with 5 previous days as input using a walk-forward validation approach on the test set.*

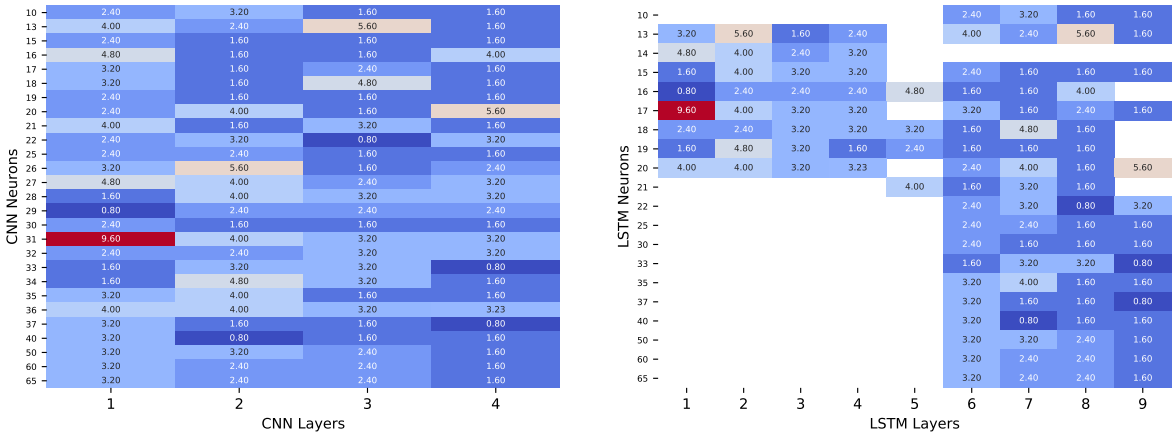


(a) *The change in accuracy of the Multi-head model whilst making predictions using a walk-forward validation approach on the test set.*



(b) *The error between the predicted and actual values of the Multi-head model whilst using a walk-forward validation approach on the test set.*

FIGURE A.177: *A graphical representation of how the accuracy and error of the Multi-head model changed whilst making predictions on the test set for test 12.*



(a) *A heatmap that illustrates the best accuracy obtained for different architectures for the CNN of the Multi-head model with 5 previous days as input.*

(b) *A heatmap that illustrates the best accuracy obtained for different architectures for the LSTM of the Multi-head model with 5 previous days as input.*

FIGURE A.178: *A graphical representation of the influence that different architectures have on prediction accuracy for test 12 of the Multi-head model.*

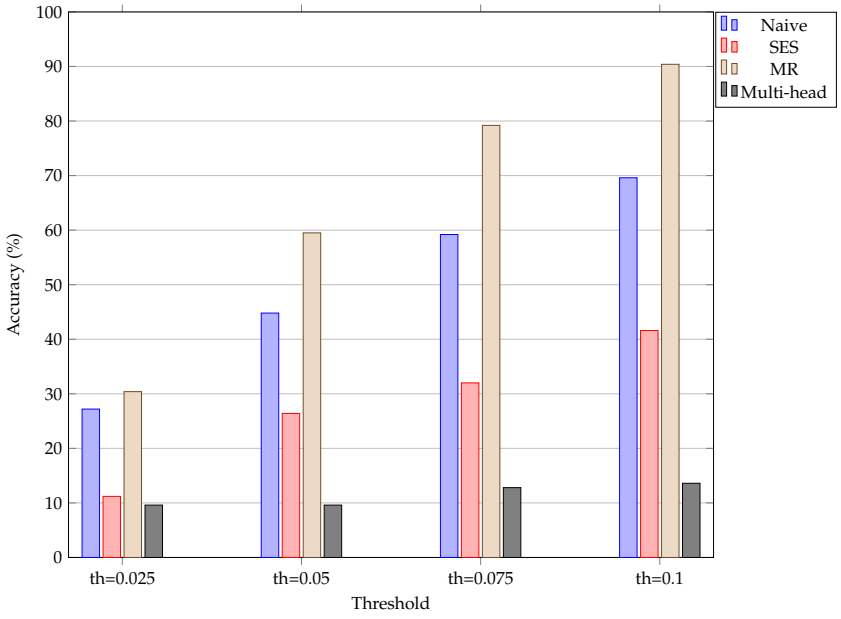


FIGURE A.179: *An illustration of how the accuracy of the different models change when the threshold used to calculate the accuracy is increased incrementally for test 12. The "th=0.025" represents a threshold set at R0.025.*

Model	Threshold of R0.025 (%)	Threshold of R0.05 (%)	Threshold of R0.075 (%)	Threshold of R0.1 (%)
Naive	27.2	44.8	59.2	69.6
SES	11.2	26.4	32.0	41.6
MR	30.4	59.5	79.2	90.4
Multi-head	9.6	9.6	12.8	13.6

TABLE A.71: *The change in prediction accuracy of the different models when the accuracy threshold in increased incrementally by R0.025 for test 12.*

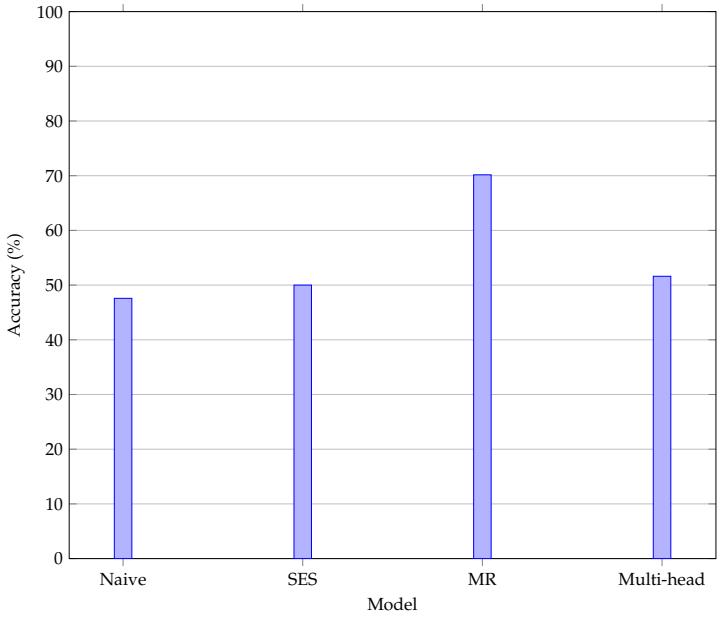


FIGURE A.180: *An illustration of the accuracy obtained when the correct prediction of price movement is considered for test 12.*

Model	Accuracy (%)
Naive	46.774
SES	50.000
MR	70.161
Multi-head	47.581

TABLE A.72: *The accuracy obtained by the different models when predicting the direction of price movement is considered for test 12.*

A.4 Input features

Table A.73 describes all the input features used for every test.

Test	# Features	Base features	Additional features
1	4	/	open, high, low, close (OHLC)
2	6	OHLC	9MA, 21MA
3	7	OHLC	positive increase (binary), negative decrease (binary), no change (binary)
4	12	OHLC	Advance, Decline, Avg Gain, Avg Loss, RS, RSI, Overbought (binary), Oversold (binary)
5	17	OHLC	HA open, HA high, HA low, HA close, Change in close, positive change, negative change, size of body, bull/bear candlestick, size of shadow, 20 SMA, 2stdev up from SMA, 2stdev down from SMA
6	24	OHLC	Tenkan-sen, Kijun-sen, Senkou Span A, Senkou Span B, Chikou Span, Diff current price and lagging line, Pos diff, Neg diff, Diff tenkan and Chikou, Pos diff (tenkan and LAG), Neg diff (tenkan and LAG), Diff Kijun and Chikou, Pos diff (Kijun and LAG), Neg diff (Kijun and LAG), Diff Span A and Chikou, Pos diff (Span A and LAG), Neg diff (Span A and LAG), Diff Span B and Chikou, Pos diff (Span B and LAG), Neg diff (Span B and LAG)
7	8	OHLC	20 SMA, 2stdev up from SMA, 2stdev down from SMA, size of bollinger band
8	11	OHLC	3 month implied volatility open, 3 month implied volatility high, 3 month implied volatility low, 3 month implied volatility close, change in 3 Month Volatility close, 3 month implied volatility positive change (binary), 3 month implied volatility negative change (binary)
9	47	OHLC	1 month implied volatility close bid, 1 month implied volatility close ask, change in 1 Month Volatility bid, 1 month implied volatility positive change (binary), 1 month implied volatility negative change (binary), 3 month implied volatility close bid, 3 month implied volatility open, 3 month implied volatility high, 3 month implied volatility low, 3 month implied volatility close, change in 3M onth Volatility close, 3 month implied volatility positive change (binary), 3 month implied volatility negative change (binary), 6 month implied volatility close bid, 6 month implied volatility close ask, change in 6 Month Volatility bid, 6 month implied volatility positive change (binary), 6 month implied volatility negative change (binary), 9 month implied volatility close bid, 9 month implied volatility close ask, change in 9 Month Volatility bid, 9 month implied volatility positive change (binary), 9 month implied volatility negative change (binary), 1 month risk reversal close bid, 1 month risk reversal close ask, change in 1 month risk reversal bid, 1 month risk reversal positive change (binary), 1 month risk reversal negative change (binary), 3 month risk reversal close bid, 3 month risk reversal close ask, change in 3 month risk reversal bid, 3 month risk reversal positive change (binary), 3 month risk reversal negative change (binary), 6 month risk reversal close bid, 6 month risk reversal close ask, change in 6 month risk reversal bid, 6 month risk reversal positive change (binary), 6 month risk reversal negative change (binary), 9 month risk reversal close bid, 9 month risk reversal close ask, change in 9 month risk reversal bid, 9 month risk reversal positive change (binary), 9 month risk reversal negative change (binary),
10	12	OHLC	Change (1st difference), Increase (binary), decrease (binary), No Change (binary) 2nd diff, increase (binary), decrease (binary), no change (binary)
11	102	OHLC	All features
12	31	OHLC	Features determined by PCA

TABLE A.73: *A table showing the input features for all the different tests that were conducted.*

Bibliography

[1] *Base SAS Software*, Available from https://support.sas.com/en/software/base-sas-support.html#documentation, [Online], [Cited February 14, 2020].

[2] *Foreign Exchange Options (FX Options) – What are they?*, Available from https://www.tradefinanceglobal.com/currency/foreign-exchange-options/#:~:text=Call%20Option%20%E2%80%93%20This%20gives%20the,up%20to%20the%20expiration%20date., [Online], [Cited June 10, 2020].

[3] *Keras: The Python Deep Learning library*, Available from https://keras.io/, [Online], [Cited February 14, 2020].

[4] *NVIDIA cuDNN GPU Accelerated Deep Learning*, Available from https://developer.nvidia.com/cudnn, [Online], [Cited February 14, 2020].

[5] *Python 3.6*, Available from https://www.python.org/downloads/release/python-360.

[6] *sklearn.preprocessing.MinMaxScaler*, Available from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html, [Online], [Cited June 10, 2020].

[7] *South Africa - Agriculture Sector*, Available from https://www.export.gov/article?id=South-Africa-agricultural-equipment, [Online], [Cited November 23, 2019].

[8] Bao W, Yue J and Rao Y, 2017, *A deep learning framework for financial time series using stacked autoencoders and long-short term memory*, PLOS ONE, **12(7)**, pp. 1–24, Available from https://doi.org/10.1371/journal.pone.0180944.

[9] Bhattacharjee J, *Some Key Machine Learning Definitions*, Available from https://medium.com/technology-nineleaps/some-key-machine-learning-definitions-b524eb6cb48, [Online], [Cited February 13, 2020].

[10] Bhattacharyya I, *Support Vector Regression Or SVR*, Available from https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff, [Online], [Cited February 4, 2020].

[11] Boginski V, Butenko S and Pardalos PM, 2006, *Mining market data: A network approach*, Computers and Operations Research, **33(11)**, pp. 3171–3184.

[12] Brownlee J, 2016, *Master Machine Learning Algorithms Discover How They Work and Implement Them From Scratch*, volume 1, Available from http://machinelearningmastery.com.

[13] Brownlee J, 2017, *Long Short-Term Memory Networks With Python*, volume 1.

[14] Cao L and Tay F, 2000, *Feature Selection for Support Vector Machines in Financial Time Series Forecasting*, **1983**, pp. 41–65.

[15] Cao LJ and Tay FEH, 2003, *Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting*, IEEE, **14(6)**, pp. 268–273.

[16] Chen J, *Foreign Exchange (Forex)*, Available from `https://www.investopedia.com/terms/f/foreign-exchange.asp`, [Online], [Cited January 22, 2020].

[17] Chen J, *Mean Reversion Definition*, Available from `https://www.investopedia.com/terms/m/meanreversion.asp`, [Online], [Cited February 6, 2020].

[18] Colby RW, 2003, *The encyclopedia of technical market indicators*, 2nd Edition, McGraw-Hill.

[19] Dash R and Dash PK, 2016, *A hybrid stock trading framework integrating technical analysis with machine learning techniques*, The Journal of Finance and Data Science, **2(1)**, pp. 42–57, Available from `http://dx.doi.org/10.1016/j.jfds.2016.03.002`.

[20] Ganti A, *Implied Volatility - IV*, Available from `https://www.investopedia.com/terms/i/iv.asp`, [Online], [Cited February 13, 2020].

[21] Géron A, 2019, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*, 1st Edition, O'Reilly Media, Sebastopol, Available from `http://shop.oreilly.com/product/0636920142874.do`.

[22] Göçken M, Özçalici M, Boru A and Dosdoğru AT, 2016, *Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction*, Expert Systems with Applications, **44**, pp. 320–331.

[23] Gollin D, Parente S and Rogerson R, 2002, *The role of agriculture in development*, American Economic Review, **92(2)**, pp. 160–164, Available from `http://www.aeaweb.org/articles?id=10.1257/000282802320189177`.

[24] Goodfellow I, Bengio Y and Courville A, 2016, *Deep Learning*, MIT Press, `http://www.deeplearningbook.org`.

[25] Hiransha M, Gopalakrishnan EA, Menon VK and Soman KP, 2018, *NSE Stock Market Prediction Using Deep-Learning Models*, Procedia Computer Science, **132(Iccids)**, pp. 1351–1362, Available from `https://doi.org/10.1016/j.procs.2018.05.050`.

[26] Hubel DH and Wiesel TN, 1963, *Shape and arrangement of columns in cat's striate cortex*, The Journal of Physiology, **165(3)**, pp. 559–568, Available from `https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1963.sp007079`.

[27] Hussain AJ, Ghazali R, Al-Jumeily D and Merabti M, 2006, *Dynamic ridge polynomial neural network for financial time series prediction*, Proceedings of the 2006 Innovations in Information Technology, pp. 1–5.

[28] Joshi P, 2017, *Artificial intelligence with Python*, 1, 1st Edition, Packt Publishing Ltd, Mumbai.

[29] Kirk M, 2017, *Thoughtful machine learning*, 1st Edition, O'Reilly Media, California.

[30] Kuepper J, *Heikin-Ashi: A Better Candlestick*, Available from `https://www.investopedia.com/trading/heikin-ashi-better-candlestick/`, [Online], [Cited February 19, 2020].

[31] Li B, Hoi SC, Sahoo D and Liu ZY, 2015, *Moving average reversion strategy for online portfolio selection*, Artificial Intelligence, **222**, pp. 104–123, Available from `http://dx.doi.org/10.1016/j.artint.2015.01.006`.

[32] McNelis PD, 2005, *Neural Networks in Finance: Gaining Predictive Edge in the Market*, 1st Edition, Elsevier Academic Press.

[33] Mitchell C, *Ichimoku Cloud Definition and Uses*, Available from `https://www.investopedia.com/terms/i/ichimoku-cloud.asp`, [Online], [Cited February 19, 2020].

[34] Nguten M, *Illustrated Guide to LSTM's and GRU's: A step by step explanation*, Available from `https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21`, [Online], [Cited April 24, 2020].

[35] Ni L, Li Y, Wang X, Zhang J, Yu J and Qi C, 2019, *Forecasting of Forex Time Series Data Based on Deep Learning*, Procedia Computer Science, **147**, pp. 647–652, Available from `https://linkinghub.elsevier.com/retrieve/pii/S1877050919302066`.

[36] O'Rourke N and Hatcher L, 1994, .

[37] Orr MJ, 1996, *Introduction to radial basis function networks*, University of Edinburgh, pp. 1–67.

[38] Osinga D, 2018, *Deep Learning Cookbook*, 1st Edition, O'Reilly Media.

[39] Prado MLD, 2018, *Advances in Financial Machine Learning*, 1st Edition, John Wiley & Sons, Inc., New Jersey.

[40] Reid D, Hussain AJ and Tawfik H, 2014, *Financial time series prediction using spiking neural networks*, PLoS ONE, **9(8)**, pp. 1–13.

[41] Ruder S, 2016, *An overview of gradient descent optimization algorithms*, CoRR, **abs/1609.04747**, Available from `http://arxiv.org/abs/1609.04747`.

[42] Schönrock-Adema J, Heijne-Penninga M, Van Hell EA and Cohen-Schotanus J, 2009, *Necessary steps in factor analysis: Enhancing validation studies of educational instruments. the PHEEM applied to clerks as an example*, Medical Teacher, **31(6)**.

[43] Sezer OB, Ozbayoglu M and Dogdu E, 2017, *A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters*, Procedia Computer Science, **114**, pp. 473–480, Available from `https://doi.org/10.1016/j.procs.2017.09.031`.

[44] Smith T, *Risk Reversal*, Available from `https://www.investopedia.com/terms/r/riskreversal.asp`, [Online], [Cited February 13, 2020].

[45] Stevenson A, 2010, *Oxford Dictionary of English*, Oxford University Press, Available from `https://www.oxfordreference.com/view/10.1093/acref/9780199571123.001.0001/acref-9780199571123`.

[46] Wilson J and Keating B, 2009, *Business Forecasting with Forecastx*, Asia Higher Education Business & Economics Management and Organization, McGraw-Hill Education, Available from `https://books.google.co.za/books?id=XXs8PgAACAAJ`.

[47] Xue J, Zhou SH, Liu Q, Liu X and Yin J, 2018, *Financial time series prediction using $\ell_{2,1}$RF-ELM*, Neurocomputing, **277**, pp. 176–186, Available from `https://doi.org/10.1016/j.neucom.2017.04.076`.

[48] Zhang B, 2018, *Foreign exchange rates forecasting with an EMD-LSTM neural networks model*, Journal of Physics: Conference Series, **1053(1)**.

[49] Zhang K, Zhong G, Dong J, Wang S and Wang Y, 2019, *Stock Market Prediction Based on Generative Adversarial Network*, Procedia Computer Science, **147**, pp. 400–406, Available from `https://linkinghub.elsevier.com/retrieve/pii/S1877050919302789`.